

AD-A033 537

PATTERN ANALYSIS AND RECOGNITION CORP ROME N Y
PROTOTYPE PAGE READER DESIGN STUDY.(U)

F/G 9/2

UNCLASSIFIED

NOV 76 P C KANE, E G FISHER, F H FENG
PAR-76-22

RADC-TR-76-325

F30602-75-C-0269
NL

1 of 2
AD-A033537



12

FL

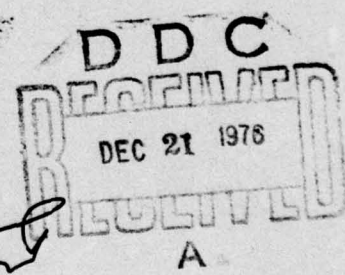
RADC-TR-76-325
Final Technical Report
November 1976



PROTOTYPE PAGE READER DESIGN STUDY

Pattern Analysis and Recognition Corporation

Approved for public release;
distribution unlimited.



ROME AIR DEVELOPMENT CENTER
AIR FORCE SYSTEMS COMMAND
GRIFFISS AIR FORCE BASE, NEW YORK 13441

ADA033537

Readers of this report are advised that some of the Sample Data included herein are relatively poor quality reproductions. They are, however, the best available.

This report has been reviewed by the RADC Information Office (OI) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public including foreign nations.

This report has been reviewed and is approved for publication.

APPROVED:

Dennis A. Nawoj

DENNIS A NAWOJ
Project Engineer

APPROVED:

Howard Davis

HOWARD DAVIS
Technical Director
Intelligence & Reconnaissance Division

FOR THE COMMANDER:

John P. Huss

JOHN P. HUSS
Acting Chief, Plans Office

Do not return this copy. Retain or destroy.

ADDITIONAL	✓
NTIS	
AVAILABILITY	
MAIL ROOM	
SPECIAL	

A

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 18 RADC-TR-76-325 ✓	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) 6 PROTOTYPE PAGE READER DESIGN STUDY ✓	5. TYPE OF REPORT & PERIOD COVERED 9 Final Technical Report June 1975 - July 1976	6. PERFORMING ORG. REPORT NUMBER 14 PAR 76-22 ✓
7. AUTHOR 12 Mr. Paul C. Kane, ↓ Dr. Edward G. Fisher, Mr. George E. Forsen Dr. Frank H. Feng, Mr. Earl W. Caldwell	8. CONTRACT OR GRANT NUMBER(s) 15 F30602-75-C-0269 NEW	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 64750F 20530205 1702
9. PERFORMING ORGANIZATION NAME AND ADDRESS Pattern Analysis and Recognition Corporation 228 W. Dominick Street Rome NY 13440 ✓	11. CONTROLLING OFFICE NAME AND ADDRESS Rome Air Development Center (IRDE) Griffiss AFB NY 13441	12. REPORT DATE November 1976
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Same	13. NUMBER OF PAGES 126 12 136p.	15. SECURITY CLASS. (of this report) UNCLASSIFIED
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report) Same		
18. SUPPLEMENTARY NOTES RADC Project Engineer: Dennis R. Nawoj (IRDE)		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Correlation Pattern Recognition Data Processing Printed Characters Microprocessors Optical Character Recognition		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This report describes the results obtained, and the methods and data used in the development and computer simulation of classification logic for an optical character reader (OCR). The work was based on that of a previous effort reported in RADC-TR-75-232. The OCR under development is to have the capability of reading unformatted non-OCR text such as might be found in foreign technical journals. Thus the classification logic has to accommodate a wide variety of print quality		

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

390 101
602

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

degradations found in "uncooperative" environments.

A high resolution, self-normalizing, correlation scheme was evaluated and found effective in handling certain types of random noise and changes in average reflectance and contrast. However, it was sensitive to systematic noise such as character rotation and stroke thickness variations.

The decision method used was to compute and choose the minimum Euclidean distance between the normalized character array and the set of normalized stored mask arrays. The masks were generated simply by registering and averaging four representative examples for each symbol in each font. The normalization was such that the mean value and standard deviation of the grey values for each array was 0, and 1, respectively. In addition, the arrays were shifted to match centroids.

Ten fonts (five Latin and five Cyrillic) of data, totalling 17,611 characters, were used in the experiments. Forced decision error rates (no rejects allowed) averaged less than 0.5% with the best being 0% and the worst 1.1%.

Most of the errors were among a small number of characters, called confusion groups. Special logic was devised for one confusion group that was common among Cyrillic fonts, and applied to the font with the worst error rate. The error rate using the special logic went from 1.1% to 0%.

In addition to the classification logic design, a scheme was evaluated for character parsing which did not require fixed pitch. Statistics differentiating test and non-text areas were devised.

The simulation included experiments aimed at providing cost-performance trade-off data for hardware design considerations. It was found that limiting the grey scale resolution to 4-bits had no significant effect on performance after normalization. It was also found that going from 40 to 80 micron spacing between image samples generally increased the error rate by a factor of two. The effect was more pronounced on Cyrillic fonts that have a preponderance of thin, but important strokes, connecting more massive, but non-discriminating strokes.

Preliminary array processor logic was proposed based on current LSI technology that would implement the design simulated. A breadboard using a minimum complement of processors should now be built to demonstrate the speed and reliability on increased amounts of data.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

TABLE OF CONTENTS

<u>Section</u>		<u>Page</u>
1.	Introduction.	1-1
2.	Procedure.	2-1
3.	Algorithms	3-1
3.1.	Previously Identified Problems	3-1
3.2.	Text Find.	3-3
3.3.	Parsing and Character Find	3-3
3.4.	Preprocessing.	3-6
3.5.	Initial Registration	3-8
3.6.	Grey-Level Normalization	3-8
3.7.	Correlation.	3-8
3.8.	Weighted Correlation	3-11
3.9.	Mask Generation.	3-11
3.10.	Summary.	3-11
4.	Experimental Results	4-1
4.1.	Text vs. Non-Text.	4-1
4.2.	Recognition Results.	4-2
4.3.	Error vs. Rejection Trade-Off.	4-6
4.4.	Analysis of 3-Phase Classifier	4-11
5.	Improvements to the Recognition Scheme . .	5-1
6.	Hardware Considerations	6-1
6.1.	Background.	6-1
6.2.	The Proposed Trial Hardware for Classifier Logic.	6-2

6.3.	Technical Approach.	6-5
6.4.	Process Time Estimate	6-6
7.	Summary and Future Work	7-1
References		R-1
Appendix A		A-1
Appendix B		B-1
Appendix C		C-1

LIST OF FIGURES

<u>Figure</u>		<u>Page</u>
3-1	The Placement of an Isolated Pattern into a Virtual Array.	3-2
3-2	The Inclusion Problem is Solved.	3-2
3-3	LIPS Image.	3-4
3-4	Grey-Level Distribution of Figure 3-3	3-4
3-5	Horizontal Line Distribution of Figure 3-3. .	3-5
3-6	Vertical Histogram of Line 2 of Figure 3-3. .	3-7
3-7	An Example of "Trimmed" or "Lost" Serifs. . .	3-10
3-8	Application of the Weighted Difference Matrix	3-12
3-9	Summary of Experimental Procedure to Test Classification Algorithms.	3-13
3-10	Summary of Classification Logic Flow	3-14
4-1	Statistics From Text Line.	4-3
4-2	Statistics From Non-Text Line.	4-4
4-3	Examples of Noise Associated with Extracted Characters.	4-10
4-4	Font C6 Error Trade-Off Analysis.	4-14
4-5	Font L8 Error Trade-Off Analysis.	4-15
5-1	Example of Character Overlap.	5-2
5-2	Example of Split Character.	5-2
5-3	V Mask, Font C2.	5-3
5-4	H Mask, Font C2.	5-3
5-5	Π Mask, Font C2.	5-4
5-6	H Sample.	5-4

5-7	H Mask, H Sample Overlay.	5-5
5-8	H Mask, H Sample Overlay Difference	5-5
5-9	Π Mask, H Sample Overlay Difference.	5-6
5-10	Π Mask, H Sample Overlay	5-6
5-11	H Sample.	5-8
5-12	H Mask, H Sample Overlay Difference	5-9
5-13	H Mask, H Sample Overlay	5-9
5-14	Π Mask, H Sample Overlay Difference.	5-10
5-15	Π Mask, H Sample Overlay	5-10
5-16	Row and Column Histograms of H Character	5-11
5-17	Row and Column Histograms of H Character	5-11
5-18	Row and Column Histograms of Π Character	5-11
6-1	Proposed Character Processor Subsystem.	6-4

LIST OF TABLES

<u>Table</u>		<u>Page</u>
4-1	Error Rates on Cyrillic Data.	4-5
4-2	Errors in Cyrillic Data.	4-7
4-3	Error Rates on Latin Data.	4-8
4-4	Errors in Latin Data.	4-9
4-5	Font C6 Error Trade-Off Analysis.	4-12
4-6	Font L8 Error Trade-Off Analysis.	4-13
4-7	Expected No. of Correlations and Frequency of Occurrence for Cyrillic Data.	4-18
4-8	Number of Characters Reaching Each Stage of Classification.	4-24
4-9	Recognition Results for Each Stage of Classifi- cation.	4-28

EVALUATION

This document is the final technical report, based on a 12 month effort, which summarized the technical efforts undertaken to develop software techniques. These techniques cover the classification and recognition of five uncooperative fonts each of Latin and Cyrillic characters which appeared in various scientific and technical journals.

This effort was a follow on to a previous contract F30602-74-C-0309 "OCR Software Development", which resulted in the evaluation of a cross correlation technique. The cross correlation technique utilizing grey scale was highly position sensitive, but relatively insensitive to random noise appearing in textual material.

The results of this effort provided information that validated the usefulness of grey scale in the recognition of thinly stroked Cyrillic characters, but de-emphasized the amount of data per character required for accurate recognition. The special software logic that was created to recognize the Latin and Cyrillic characters along with the confusion characters in each group achieved an error rate of approximately 0.5% on the unformatted text. However, the speed of recognition for an actual system is far too slow in software. The hardware recommendations in section 6 of this report are an initial attempt to demonstrate the advantages of utilizing microprocessor technology for character recognition and employ parallel processing techniques to significantly reduce the per character processing time. In a future OCR device that would take advantage of these techniques, decisions must be made regarding the cost effectiveness of having a small number of errors vs the processing time trade off.

Dennis R. Nawoj
DENNIS R. NAWOJ
Project Engineer

SECTION 1

INTRODUCTION

✓ The past fifteen years have produced significant achievements in the field of Optical Character Recognition (OCR). Today there exist a number of commercially-available OCR systems which are capable of reading a wide variety of type fonts, provided that the input is composed of "high quality" material prepared in a cooperative environment. In terms of human recognition, the restrictions implied by the term "high quality" material must be considered severe. The machine reject rates on degraded material simply do not correspond to those of the human.

To some degree, the performance differential between man and machine is attributable to the machine's incomplete use of available information. For reasons of economy, commercially-available OCR systems tend to use relatively simplistic recognition strategies involving only a few grey levels. In lieu of increasing the recognition computation and therefore the cost, the commercial manufacturers have chosen to place the burden on the user to produce "high quality" material at the data source origin. This procedure can work well when the OCR system is placed in a cooperative environment where the source data are specially prepared for OCR processing. However, the scanning needs of the Air Force often do not lend themselves to such cooperative environments. In some OCR applications, absolutely no control over the source generation is possible, and yet the desire to reduce media input costs still exists. One such application is the requirement to encode the text of foreign journals for the purpose of automatic translation and information retrieval and analysis. This material is quite varied in both quality and format, so much so that present-day commercially-available OCR systems have been judged inadequate.

✓ This report describes the results obtained on a research project to evaluate a recognition technique for typeset Latin (English) and Cyrillic (Russian) alphabetic characters which have been scanned and digitized at comparatively high spatial resolution and high grey-level quantization by the LIPS scanner*. A normalized correlation technique was employed as the recognition algorithm, at the suggestion of the Government. This technique allows recognition in severe noise conditions such as might occur with poor quality printing. ✕

Several problems inherent in correlation techniques were addressed. These problems were demonstrated during work reported in the Final Technical

* RADC developed a high-resolution laser scanner system known as the Laser Image Processing Scanner (LIPS). The LIPS system is capable of scanning, digitizing, and storing on magnetic tape, source material with each resolvable point represented by one of 256 grey levels, at sample spot size and spacings from 1.25 to 40 microns.

Report (RADC-TR-75-232) for a related contract (F30602-74-C-0309). Basically, the solution to the problems required finding the best possible registration of the scanned character to the prototype mask so that spurious correlations with the wrong mask would not lead to incorrect classification, and by preclassifying by size so to reduce the number of candidate classes per character. Additional logic was found necessary to resolve certain confusion groups.

A correlation technique was applied as the character recognition algorithm for five Latin and five Cyrillic fonts, using data with high spatial and grey-scale resolution. A data base of characters from the above mentioned fonts was collected from selected Russian technical journals and digitized using the LIPS scanner at a spot size of 40 microns, using 256 grey levels. These data were chosen to be most representative given the constraints imposed by time and scanning and program processing rates. Initial investigations revealed that the data could be reduced by simple truncation to 64 grey levels without any influence upon the results. This could be done to reduce overall storage needs.

A character isolation algorithm was also developed to segment the digitized images first into lines, then to words and then to individual characters. The isolated characters were then identified.

Several examples of each class were selected to form prototype masks while the remainder were set aside to form a test set. After some preprocessing, designed to enhance the overall system performance, correlation was performed. Statistics on the correlation coefficients as a function of character class and font were obtained to measure the effectiveness of the correlation technique. Font-by-font statistics were gathered to determine both character class-pair problems and font characteristics which cause problems.

Over seventeen thousand characters were isolated and classified in this study. The overall accuracy rate was better than 99.5%. The results were about as expected. The additional resolution allowed a high level of discrimination for perfectly-aligned, well formed character-mask pairs. In other words, the character-mask variance* was generally quite small for a test character and the corresponding stored mask--several times smaller than the minimum variance between a character and most other masks.

Correlation is an area-sensitive technique. Thus any degradation that alters the area of overlap between a character and its correct mask will seriously degrade their correlation. Random noise does not have as strong an effect on performance as does a rigid translation or rotation, until the noise is of such magnitude as to affect the same percentage of picture elements (pixels) as would the misregistration. Thus a registration algorithm was required. The scheme employed matched centroids and then made four small x and y displacements computing a maximum of nine correlations per character-mask pair.

* As measured by the Euclidean distance between normalized character and mask vectors.

Additional recommendations for modification of the recognition scheme were made to alleviate other problems generally associated with correlation. Initially it was thought necessary to include in the variance calculation a weighting vector which gives emphasis to those areas that are critical for discriminating between similar characters. This is especially true for certain Cyrillic characters which differ only in a small fraction of the total area, usually in the thin connecting strokes. Thus an alternative method was tried which proved successful on a small sample of confusion classes. A scheme was tried that weighted areas of difference between masks. However, unwanted differences - usually due to variation in stroke width - were often as large, in area, as the important areas of difference.

Section 2 describes the systems, procedures, and programs used in obtaining data, masks, and processed results.

Section 3 describes the algorithms and procedures in greater detail.

Section 4 details the experimental results.

Section 5 further discusses suggested embellishments to the basic scheme.

Section 6 describes the effort to design test hardware for the classification logic.

Section 7 summarizes the current work and outlines suggested future work.

Appendix A provides the software documentation.

Appendix B provides the experimental design timing estimates of a single processor classification scheme.

Appendix C includes pages of test extracted from each of the Russian journals used.

SECTION 2

PROCEDURE

The systems, programs, and procedures used in the operational steps of the project are discussed in this section. The data were scanned on the LIPS system. For most of the project, the scanned data had to be copied from the nine-track tapes output by LIPS onto seven-track tapes for general use on the HIS-6180 MULTICS system. Several programs were developed on the MULTICS system for evaluating the correlation algorithms. Some of the image data processed were taken from data scanned under contract F30602-74-C-0309. As needed, additional images were scanned on LIPS.

The LIPS tape images were then stored as disk images by the MULTICS program OCR.

The disk files were then processed by the MULTICS program DOCR* which isolated the individual characters in the images. These characters were then displayed in groups on a CRT screen and hardcopies of the displays were made so that the characters could be visually identified. The character identities were then recorded and typed as ID files for use by a program which selects characters for building masks and another program which merges the ID file with a character file to form an edited file of labelled characters. Some characters, such as italics, bold face and mathematical symbols, were specifically labelled to be ignored by the recognition algorithm.

After all of the characters of a font were identified, the program MASK_SELECT randomly selects patterns to be used as masks and outputs the file names and indices of the patterns for use by COLLECT_PATTERNS. COLLECT_PATTERNS copies selected patterns from a set of files output by DOCR and writes them into another file. In this case it is used to collect four samples of each pattern into a file called MASK_SET. The program MASK_GENERATOR is then used to input groups of patterns, perform whatever preparation may be necessary, register them to each other, compute the "average" of the patterns, and output the resulting patterns as masks.

The entire collection of masks is then input to TRANSFORMER. TRANSFORMER then computes the grey-level normalized and truncated version of each mask and outputs it to a MASK_DIRECTORY, along with a set of pointers sufficient for CORREL_MAIN, the central correlation and classification routine, to find each mask when needed. TRANSFORMER is also applied to each of the pattern files output by DOCR; it converts each of these patterns to grey-level normalized and truncated form.

Either as DOCR files or as TRANSFORMER files, the pattern files are merged with the ID files so that CORREL_MAIN may evaluate the validity of its decision.

* For clarity, program names will be presented in upper case letters. In Appendix A, as is customary in MULTICS, they are in lower case letters.

The files of grey-level normalized, truncated, and labelled patterns are then input to CORREL_MAIN for processing with the several correlation-related algorithms. Because there are several files of patterns per font, the outputs of each run of CORREL_MAIN are merged and later processed by SUMMARIZE which prints out summary error and correct recognition rates. They may also be processed by SUMMARIZE_TRADE_OFFS which prints out sets of error, reject, and correct recognition rates for various confidence levels.

SECTION 3

ALGORITHMS

3.1. PREVIOUSLY IDENTIFIED PROBLEMS

In the preceding report [1], some problems were identified as being directly related to the use of correlation. The algorithms which applied the correlation techniques first addressed these problems. The problems identified were: (1) how to deal with the fact that some characters are included in others as parts, and therefore correlate well with those parts of the other characters; (2) how to register mask and character in order to obtain an optimum correlation; and (3) how to increase the sensitivity of correlation to particular, high-information parts of characters while reducing sensitivity to others.

The "inclusion problem" is solved by defining each actual pattern array to exist within a larger virtual pattern array. Thus, when a character pattern is isolated from its surrounding text, a larger, virtual pattern is defined as in Figure 3-1 with some fixed intensity value defined to be the additional background. The size of the "virtual" pattern is chosen to be a little larger than necessary to enclose the largest mask in the system. This does not have a significant effect on the total amount of computation necessary to compute the correlation. In this manner, the correlation algorithm is forced to compute the variance between both the included and non-included parts of test characters and the masks. Figure 3-2 contrasts the use of the virtual array technique to the earlier work in which inclusion was a problem. Size normalization would produce a similar result. However, size normalization requires additional computation for both the normalization and correlation processes, while the "virtual pattern" technique requires little additional computation.

The registration problem is essentially solved by registering the centroids (centers of mass) of the character and each mask. Observation of a set of correlations obtained by correlation of 300 well-formed characters to their own masks has shown that optimum correlation is always obtained when the character center of gravity is within ± 1 pixel in either direction of the mask centroid.

Spurious, incorrect classifications are often made unless the best registration is obtained. As a first approximation to optimum registration, the centers of mass of the character and mask were registered before the correlation was computed. With a set of 300 characters we computed all correlations for a variety of registrations of the characters to their own (correct) mask. The center of mass of the characters was registered to each of the points within ± 2 pixels in either direction of the mask centroid and each of the 25-character-to-mask correlations was performed. In all of the cases, shifting ± 1 was sufficient to obtain the optimum correlation. Often, no shifting was required. For some characters, although a shift would produce the optimum correlation, the suboptimum, unshifted correlation to one mask is so much



Isolated Pattern



Virtual Pattern Array

Figure 3-1 The Placement of an Isolated Pattern into a Virtual Array



Mask



Character



i



ii

Figure 3-2

- i) When the correlation is performed "in-text", inclusion results
- ii) When character and mask are centered in a larger, virtual array, the inclusion problem is resolved

better than any correlation to any other mask that a decision can safely and correctly be made without additional computation to find the optimum correlation. Certain characters, however, still require precise registration.

An algorithm was designed to increase the sensitivity of correlation to particular, high-information parts of characters. This algorithm will be discussed more thoroughly in Section 3.8. The following relates to algorithms used in this effort. Flow charts in Section 3.10. summarize the overall experimental recognition procedures, and the recognition logic flow.

3.2. TEXT FIND

To separate text from non-text, some statistics of various text and non-text images were studied. The statistics studied were entropy, average intensity and average length of black/white segments in the binary quantized version of the image. It was observed that each of the three statistics is reasonably consistent over any segment of either text or non-text. However, for any transition between text and non-text, at least one or possibly all of the statistics shows a significant change. The particular statistic(s) which change depends upon the specific nature of the non-text and text. No attempt was made to automate the process of separating text from non-text.

3.3. PARSING AND CHARACTER FIND

The automatic method for isolating characters in text involves the three stages of line find, word find, and character find. It is assumed that correction of page skew could be performed at this point in the processing.

The character extraction is performed by the program DOCR.* Various display functions are optionally provided by subroutines.

Initially, a histogram of pixel intensities is made. Figure 3-3 is a sample data chip as digitized by LIPS and Figure 3-4 is the grey-level histogram of the chip. From this distribution, an approximate threshold value for determining "character" vs. "background" can be selected. Figure 3-5 is composed of two distributions: the first is the background grey-levels and the second is the character grey-levels. The image is thresholded at the approximate point where the background distribution ends, with every pixel less than the threshold value set to 0. This thresholding eases the character extraction process.

A frequency count of the number of non-zero pixels in each row of the image is obtained. Even though there are several characters which rise above the height of the small lower case characters or drop below the line, this distribution is characterized by very sharp jumps that indicate the location of the main body of text lines with each line generally being separated by at least five pixels. Figure 3-5 is a histogram of the number of non-zero pixels in each row of the text shown in Figure 3-3.

* For clarity, program names will be presented in upper case letters. In Appendix A, as is customary with MULTICS, they are shown in lower case letters.

The excitation and propagation of sound perturbations, striction forces and heating of the substance during unfocused and multifocus structures are investigated theoretically. The density of the sound energy in the medium are investigated. It is shown that two perturbation mechanisms in the medium the energy produced are very different. Conditions for which variation of the focal region is close to a quasistatic one, are established. It is found that in both cases an anomalous variation region occurs when the focus velocity is close to that of heating and striction to nonlinear polarization of the medium considered is estimated.

Figure 3-3 LIPS Image

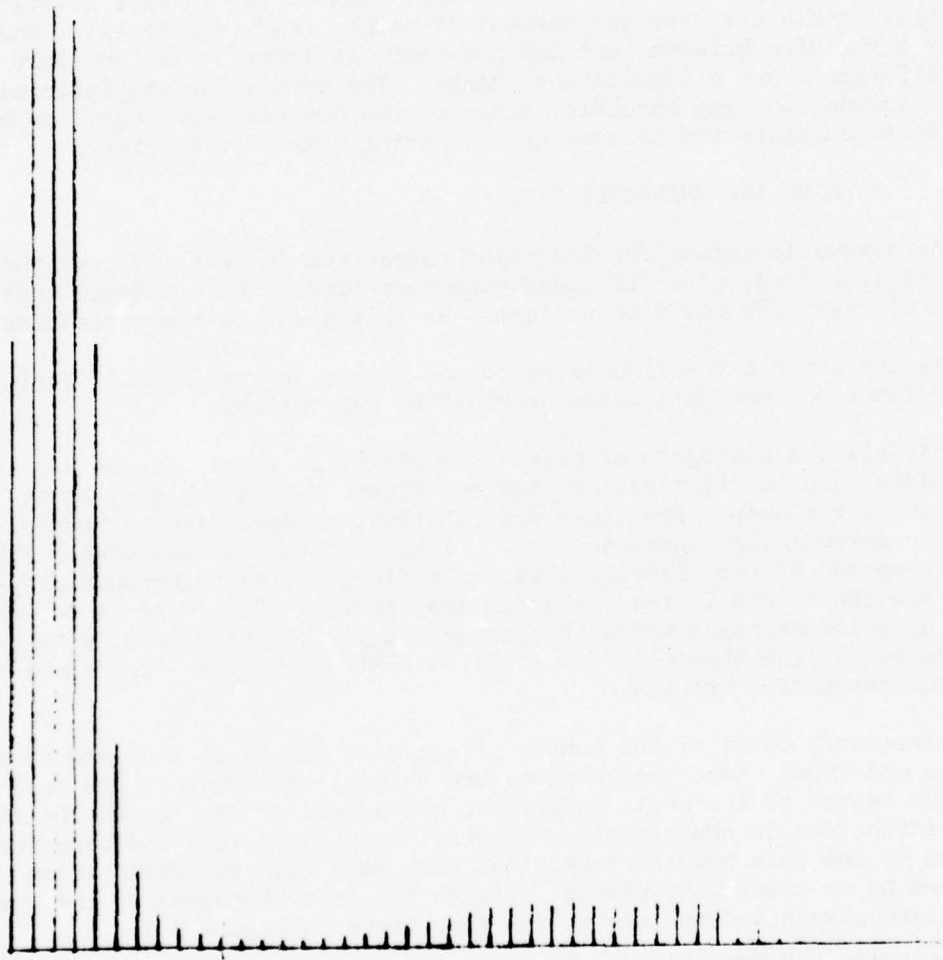


Figure 3-4 Grey-Level Distribution of Figure 3-3

Figure 3-5 Horizontal Line Distribution of Figure 3-3



The problem of correcting page skew was not a concern in this effort; however, it could be not efficiently recognized and corrected at this point in the total classification procedure. Improper skew correction would be indicated by the histogram of non-zero pixels not being sharply defined. Correction of skew could be obtained for an entire page of characters and no modifications of later algorithms would have to be performed to account for skew.

After separating lines of text, vertical or column histograms of the number of non-zero pixels are computed for each line. Figure 3-6 presents a histogram of the intensity values of line 2 of Figure 3-4. The start of a word is defined as that point in a line of text where there are at least 12 of 15 non-zero elements in that line's column histogram. Similarly, a word stop is that point where at least 12 of 15 elements are zero in the line's column histogram. When the word's boundaries are determined, the character extraction occurs.

The average character, in the fonts that this report has studied, is less than 90 pixels in width. Using this information, DOCR examines the column histogram elements of each word and locates a minimum about every 90 pixels. The distance between the start of a word and the first minimum is the width of the first character. The distance between the first and second minimums is the width of the second character, etc. A raw histogram is then computed over the width of each character to determine the height of the character. Because of character overlap, the minimum histogram element is searched for instead of a zero histogram element.

3.4. PREPROCESSING

After the characters are isolated a few preprocessing operations must be performed. The purpose of the preprocessing is to prepare the character images for the correlation programs.

Two of the preprocessing functions are designed to enhance the quality of the images and thereby improve the performance of the OCR system. The functions, SPOT-REMOVER and NOISE-CLIPPER, remove some "salt and pepper" background noise from the character patterns. The removal is performed by setting the pixels to the background fixed value.

SPOT-REMOVER scans the character pattern to find isolated spots of small density and removes them. It was sufficient to define "isolated" to mean that the spot should be separated from the primary image by at least one empty row (or column) of background. It was also sufficient to define as "small" any spot whose black pixels comprised fewer than 4% of the black pixels of the image.

NOISE-CLIPPER removes isolated black pixels. For this purpose a point is isolated if it has less than two neighbors in a four-connected sense. This function serves to improve some of the imagery since it does smooth out edges. However, it also deletes some very fine connectors in poor images and some

Figure 3-6 Vertical Histogram of Line 2 of Figure 3-3



fine, thin serifs in a few fonts. It may be necessary to bypass this option for certain fonts in certain text sources.

3.5. INITIAL REGISTRATION

The character's center of mass is then computed and the pattern is transposed so that its center of mass is at the center of the pattern. The center of mass point (\bar{x}, \bar{y}) is computed as:

$$\bar{x} = \frac{\sum_{x,y} xf(x,y)}{\sum_{x,y} f(x,y)} \quad \bar{y} = \frac{\sum_{x,y} yf(x,y)}{\sum_{x,y} f(x,y)} \quad (1)$$

where $\sum_{x,y}$ is computed for each point (x,y) within the tightest boundary around the pattern and $f(x,y)$ is the intensity value at (x,y) . IBM has reported that normalization by first and second moments was more successful for character recognition than other normalization techniques (such as size normalization) (IBM Research Report 140-68). Our centroid normalization is a normalization by first order moment. The characters are all of the same font so second order moment normalization is unnecessary.

3.6. GREY-LEVEL NORMALIZATION

This procedure converts the 6-bit, centered, and noise-detected pattern to a grey-level normalized truncated and thresholded pattern of n bits. In this operation 4 bits were used. The following correlation formula was used:

$$c = 1/N \sum_{i,j} \left| \frac{x - \bar{x}}{\sigma_x} - \frac{r - \bar{r}}{\sigma_r} \right|^2, \quad (2)$$

The normalized grey-level variable $\tilde{x} = \frac{x - \bar{x}}{\sigma_x}$ is first computed for each test character pixel. The characteristics of the variable \tilde{x} are that it has mean 0 and standard deviation 1. More important is that its range is limited.

At this point the variate \tilde{x} is a real variable which requires more than 6 bits for representation. Since the goal of this effort is to evaluate the performance of the technique for performance on a hardware OCR device, \tilde{x} is scaled and truncated by applying the transformation

$$\tilde{\tilde{x}} = 2^m \cdot \begin{cases} \min(\tilde{x}, 1.99) & \text{if } \tilde{x} \geq 0 \\ \max(\tilde{x}, -1.99) & \text{if } \tilde{x} < 0 \end{cases} \quad (3)$$

where m is the desired number of bits to be maintained from the right of the binary (or radix) point. Thence forward, the value \tilde{x} is treated as an integer.

3.7. CORRELATION

The correlation routines are a series of subprograms designed to determine the identity of a character by comparing the character pattern array to a set of masks. The comparison process requires a maximum of three passes. In the first pass the pattern is compared to all masks which meet some criterion of plausibility. We have used relative size as the criterion. If the pattern is of size $m \times n$, (m rows and n columns), each mask which is to be compared to the character must be of size $p \times q$ where $m' \leq p \leq m''$ and $n' \leq q \leq n''$. Determination of the functions which determine the relationships between m' , m'' , and m and n' , n'' , and n is font-dependent. If the serifs of the font are large and not likely to disappear because of poor ink quality or preprocessing, then $m' = .86m$, $m'' = 1.14m$, $n' = .86n$, and $n'' = 1.14n$ were found to be adequate. These parameters are sufficient to account for variations in inking, quantization error, and minor errors by the isolation process.

If the font has small, thin serifs which occasionally disappear, it is necessary to modify the definition of n'' slightly. The following is an example of the modification which had to be used:

```
n'' = 1.14 x n;  
if n'' < n + 10 then n'' = n + 10.
```

An example of the case in which this is necessary is shown in Figure 3-7.

After the first pass in which correlation is performed against all masks in the proper size range, a list of the most likely identities and their correlation of the input character is obtained. The list is then sorted so that the choice with the best correlation is first.

Depending on the system's confidence in the best choice as the proper decision, the system either accepts it as the decision or goes on to the next phase of processing.

The criterion for confidence in the decision is the ratio between the second best and the best correlations. For some fonts it was found that a ratio of less than 1.6 was sufficient, for others a ratio of 2.5 was required. It must be pointed out that the smaller the ratio that is used, the greater the probability of accepting an incorrect decision.

In the second phase of processing, the character is compared to the set of masks whose correlation (from phase 1) is less than μ times the best correlation obtained. This procedure shifts the mask ± 1 in each direction and correlates the shifted mask to the character. For each mask the best of the eight correlations thus obtained and the correlation obtained in Phase 1 is used as the character-to-mask correlation output by this phase. A faster version would use an approximate scheme requiring only four shifts.



Character



Mask

Figure 3-7 An example of "trimmed" or "lost" serifs.

Again, at the end of Phase 2 a determination is made whether to make the decision based on the best correlation or to go on to the third phase.

Before entering Phase 3, the list of candidate choices is again shortened in a manner similar to that used at the beginning of Phase 2.

3.8. WEIGHTED CORRELATION

In Phase 3, a series of pairwise comparisons is made. A pairwise comparison is performed for each pair of candidate choices remaining. Thus, if the remaining choices are c, e, and o, three pairwise comparisons are made: c vs. e, c vs. o, and e vs. o. Then the candidate with the most votes (a candidate receives a vote if it wins a comparison) is selected as the decision. The comparison technique uses a "weighted" mask which is computed as the difference of two masks and then incorporated into the correlation formula. Figure 3-8 graphically illustrates the strategy employed.

What the technique attempts to do is to build a specialized mask which represents the difference of the two masks being compared. Then, only those character-to-mask differences which occur in the non-zero portions of the weighted mask contribute to the correlation sum.

3.9. MASK GENERATION

For the previous contract, masks were generated by manually co-registering and averaging together characters using the DICIFER system. The method is both tedious and prone to operational error. Most of the masks generated for the work under this contract were generated by automatic registration and averaging. The automatic registration was performed by matching the centroids of the characters to each other and then computing the average. The masks were then subjected to the same preprocessing transformations which are applied to the test characters and stored in a mask directory for quick reference by the correlation program.

3.10. SUMMARY

The flow chart in Figure 3-9 summarizes the overall experimental recognition procedure and Figure 3-10 describes the recognition logic flow.

e

Mask₁

0

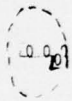
Mask₂

7

Weighted
Mask

e

Character



Weighted Difference Matrix
times Character and Mask₁



Weighted Difference Matrix
times Character and Mask₂

Figure 3-8 Application of the Weighted Difference Matrix

Figure 3-9 Summary of Experimental Procedure to Test Classification Algorithms

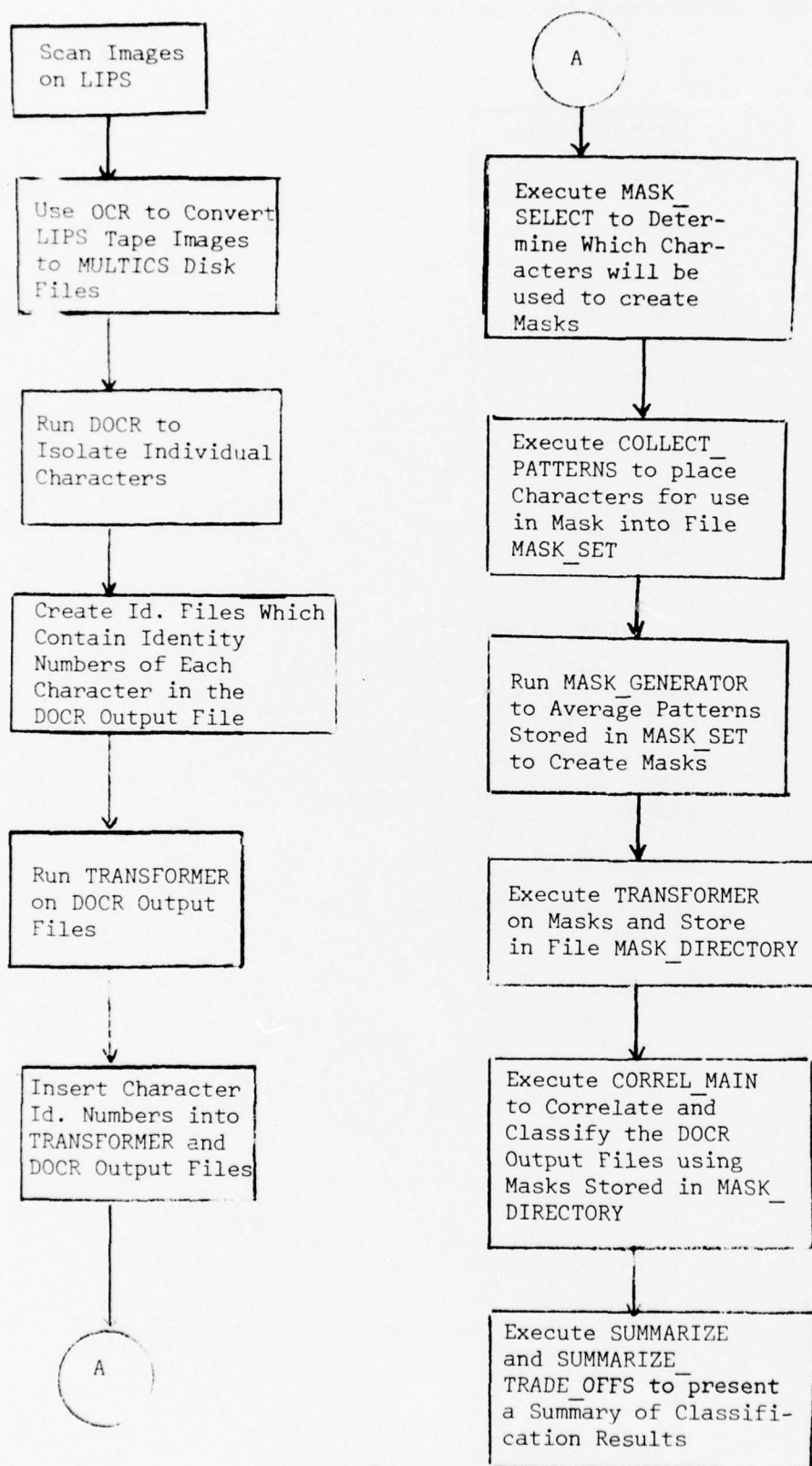


Figure 3-10 Summary of Classification Logic Flow

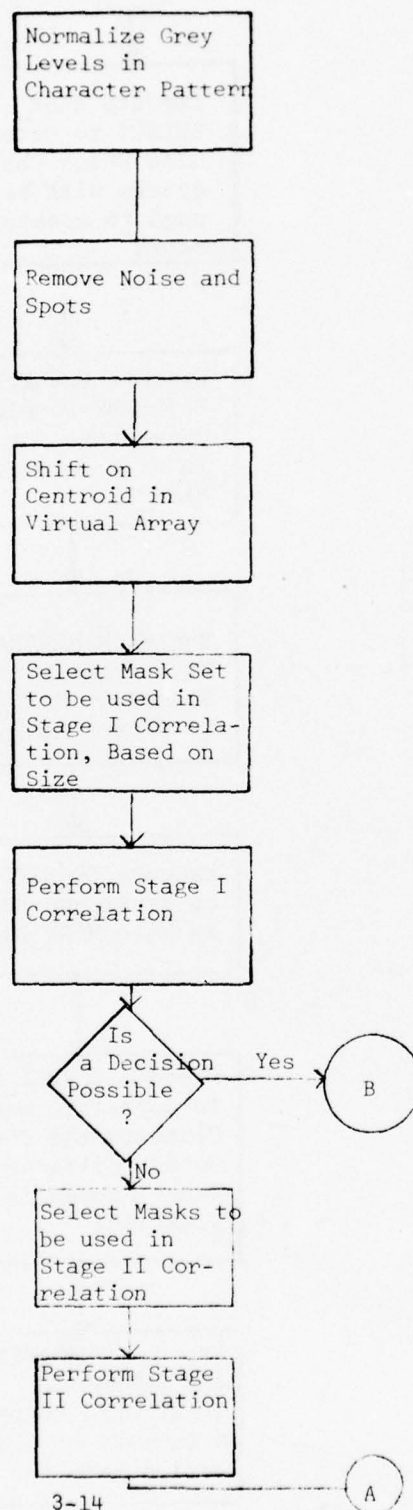
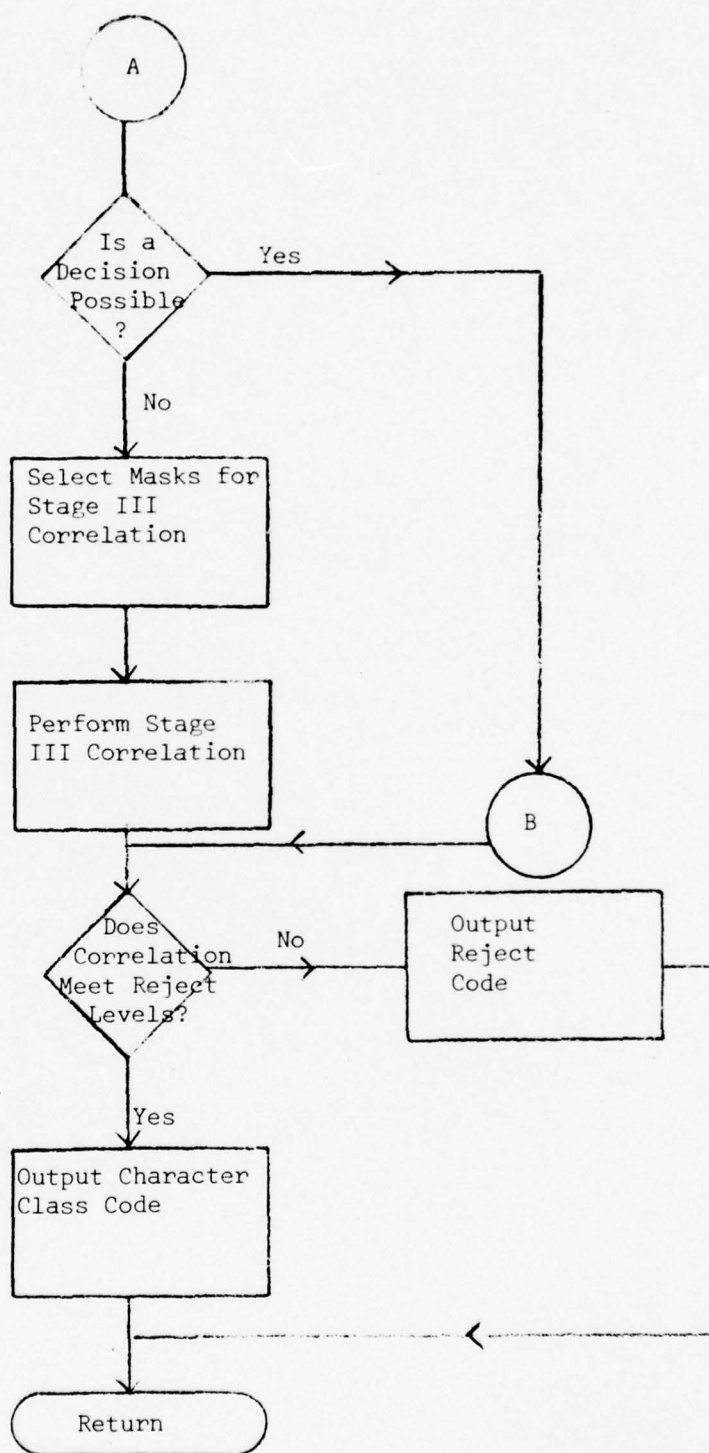


Figure 3-10 Summary of Classification Logic Flow (Cont'd)



SECTION 4

EXPERIMENTAL RESULTS

Experiments were conducted to evaluate the hardware simulation algorithms for recognition of lower case characters from 5 Latin fonts and 5 Cyrillic fonts. Error rates generally varied from 0.0 to 1.0%. For some fonts the experiments were simply a matter of running the programs which had already been developed; for others, it was an iterative process of determining where and why errors occurred, changing the mask set to compensate, and re-running the programs.

In this section, the recognition/error rates for each of the fonts are presented as well as a discussion of these errors. The results of investigations into determination of text/non-text data, error vs. rejection trade-offs, and effects of each segment of the three-pass classifier will be presented.

4.1. TEXT VS. NON-TEXT

To separate text from non-text, some statistics of various text and non-text images were studied. The statistics studied were entropy, average intensity, and average length of black/white segments in the binary quantized version of the image.

Given an M by N image $f(i,j)$, $1 \leq f(i,j) \leq K$, the entropy E , a measure of grey-level randomness, is defined as

$$E = - \sum_{k=1}^K P(k) \log_2 P(k)$$

where $P(k) = (\text{No. pixels of intensity } k)/MN$

The average intensity \bar{f} is defined as

$$\bar{f} = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N f(i,j)$$

and the average segment length \bar{s} is defined as

$$\bar{s} = MN/N_s$$

where N_s is the total number of segments in the image.

A "segment" of length s is a 1 by s subimage in which either all $f(i,j) \geq T$ or all $f(i,j) < T$ for a given threshold intensity T .

In our experiment, $M=1$, $N=500$, and $T=f$. That is, an image is simply a scan line of 500 pixels. 100 consecutive scan lines were used to examine the behavior of these statistics. Figure 4-1 shows such curves from a text image and 4-2 shows those from a non-text.

Note that each of the three statistics behaves reasonably consistent over any image of either text or non-text. However, significant changes occur within the transition area. Whether any of these three statistics is redundant and whether these statistics are sufficient to distinguish text from various non-text content require further investigation.

4.2. RECOGNITION RESULTS

Table 4-1 presents the details concerning the number of characters processed and recognized in each of the Cyrillic fonts. The percentages presented are estimates of the error rates which should be expected from reading the appropriate journals with the same techniques. The error rate is computed as

$$e = \sum_{j=1}^{33} e_j f_j$$

where the summation is over all classes j of characters, e_j is the measured fraction of errors on class j , and f_j is the frequency of class j expressed as a fraction of unity and measured over a large population. The simpler form

$$\frac{\text{number of errors}}{\text{number of sample characters}}$$

might introduce bias due to sampling an atypical distribution of characters. The frequencies of occurrence used in this study were obtained from a previous study*.

* Engineering Analysis and Digital Simulation of the Optical Russian Print Reader, Technical Documentary Report No. RADG-TDR-62-472, Sept. 3, 1962.

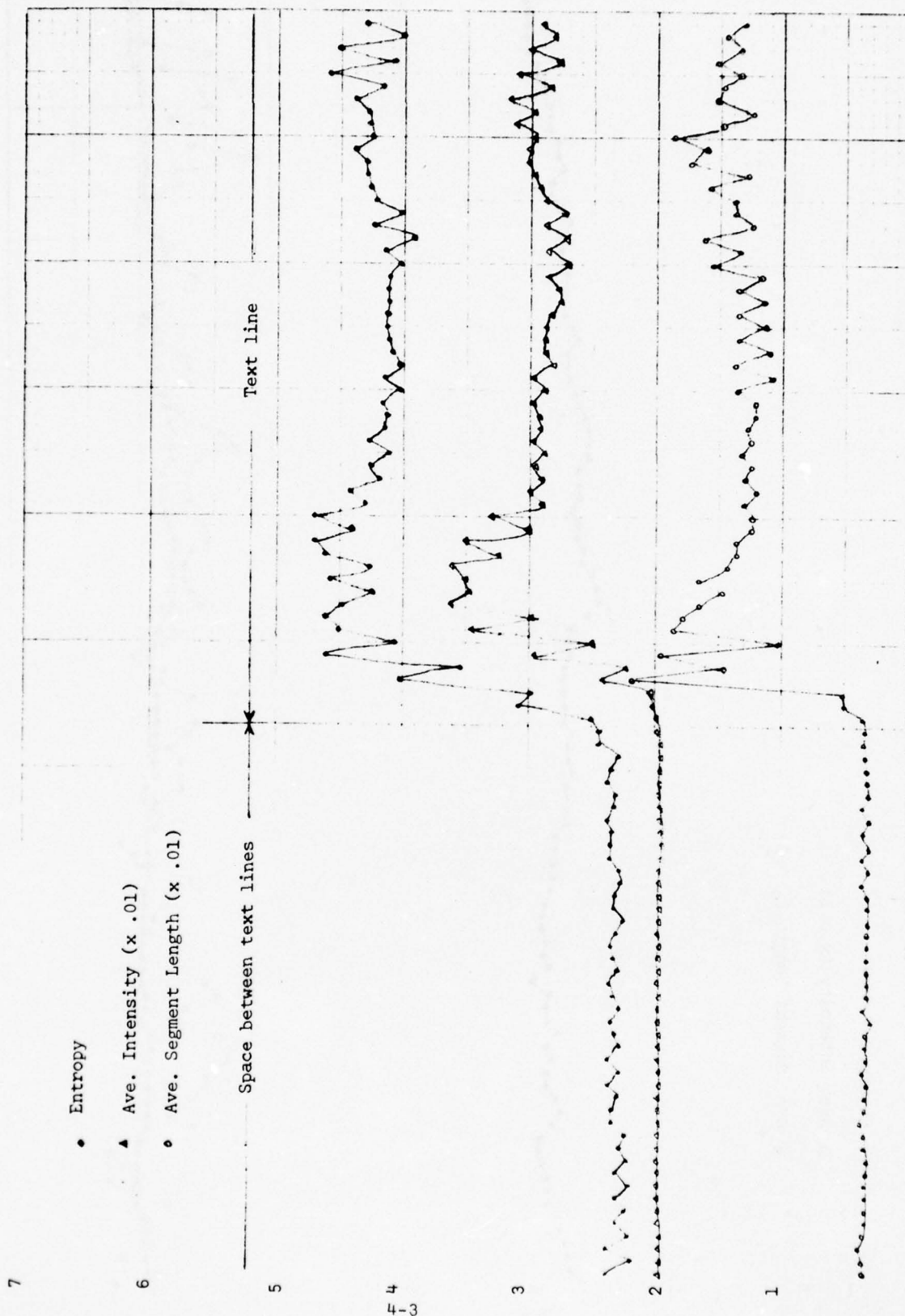


Figure 4-1 Statistics From Text Line

7

- Entropy
- △ Ave. intensity ($\times 0.1$)
- Ave. Segment Length ($\times 0.1$)

6

5



4

3

2



Figure 4-2 Statistics From Non-Text Line

Font	Number of Characteristics	Number of Errors	Error Rate	Recognition Rate
C1	1654	13	.75%	99.25%
C2	2038	9	.47%	99.53%
C4	1972	0	.00%	100.00%
C6	3085	18	.58%	99.42%
C8	1506	17	1.11%	98.89%
Total	10255	57	.57%	99.43%

Table 4-1 Error Rates On Cyrillic Data

Table 4-2 describes the problem pairs encountered in each of the Cyrillic fonts and their relative frequency of occurrence within that font. Tables 4-3 and 4-4 provide analogous information for the Latin fonts.

In the Cyrillic fonts, almost 85% of the errors (48/57) occurred among the *И*, *Н*, and *П* characters. The main difference among these three characters is the presence or absence of a cross-piece between the two strong vertical bars. (This is illustrated in more detail in Section 5.1. Another level of classification logic to deal with this confusion group is discussed in Section 5.2.) Noise (See Figure 4-3), and the varying width of this cross-piece are the two main reasons for misclassification. Other errors occurred when the stroke width of the crosspiece of "e" was thinner or thicker than that of the mask, resulting in a classification as an "o" or a "c".

In the Latin fonts, the recognition rate was generally higher than in the Cyrillic fonts, 99.74% vs. 99.43%. The main source of errors was chopped serifs. The missing serifs, probably a result of overlap of characters encountered when the characters were isolated and/or misprints, explain the *b* → *h*, *h* → *b*, *n* → *o*, *m* → *o*, *t* → *f*, *l* → *f* errors. Varying widths of the "e" crosspiece explain the *e* → *o* and *c* → *e* errors.

An experiment to determine the effect on the recognition system of a change in resolution was conducted on a subset of the C6 font. A decrease in resolution was simulated by a 50% linear decimation in each direction. This was accomplished by replacing every two pixels with the average of those two pixels. The 50% decimation also accomplished an overall storage reduction of 4:1.

There occurred 12 errors in the 1267 samples yielding a recognition rate of 99.05%. Using the full-sized patterns, the recognition rate was 99.39%. The trade-off is a reduction in the recognition rate of .34% for a 4:1 reduction in storage requirements. Considering the amount of disk space necessary for the isolated characters of font C6, 910K for approximately 3100 characters, the 4:1 reduction seemed worthwhile indeed. However, in a specialized hardware OCR system, all the isolated characters of a font would not be stored at once. Rather, they would be processed and then the storage locations released for other characters. The main hardware benefit would be reduction in processing time.

4.3. ERROR VS. REJECTION TRADE-OFF

For those fonts which had non-trivial error rates (at least 1/2 % and at least 10 misclassified characters), an analysis was made of the desirability of rejecting characters instead of making forced decisions. The strategy selected to determine the confidence in a decision was to require a particular minimum ratio of correlation before permitting a decision to be made. For example, if the correlation (actually Euclidean distance) of a character to mask "c" is 10,000 while its correlation to mask "o" is 10,010, a forced decision would decide "c". However, a decision which required a second to

Table 4-2 Errors in Cyrillic Data

font	C1	C2	C4	C6	C8
No. of	2 e → c	4 И → П		5 B → b	10 И → H
Errors	1 T → r	5 H → П		12 H → П	4 И → П
(Classified -	1 П → N			1 e → o	3 П → H
→ True	2 П → H				
Character)	5 И → H				
	1 H → И				
	1 И → П				
Total No. of Errors	13	9	0	18	17

Table 4-3 Error Rates on Latin Data

Font	No. of Characters	No. of Errors	Error Rate	Recognition Rate
L1	1599	1	.05%	99.95%
L2	1128	3	.16%	99.84%
L4	1491	2	.09%	99.91%
L5	1396	1	.08%	99.92%
L8	1742	16	.94%	99.06%
Total	7356	23	.26%	99.74%

Table 4-4 Errors in Latin Data

Font	L1	L2	L4	L5	L8
Errors (Classified → True Character)	1 L → i	2 t → f 1 L → f	2 f → t	1 c → e	2 b → h 2 h → b 1 e → o 5 n → o 6 m → o
Total No. Errors	1	3	2	1	16

е н с и в н о с т и н е с у

Figure 4-3 Examples of Noise Associated with Extracted Characters
(Tick marks denote starting columns of each character.)

first ratio of 1.01 would reject the character.

Tables 4-5 and 4-6 present sample recognition-error-reject rates for some of the fonts. Figures 4-4 and 4-5 present this relationship graphically.

For both fonts, C6 and L8, the recognition rate declined as a function of the minimum ratio required for a decision increased. Although the error rate in both instances dropped as the decision ratio increased, the overall recognition rate also decreased. This is expected, however, as the increase in the number of rejects illustrates. While the decrease in recognition rate as the ratio increased from a forced (1.00) decision to a 1.03 decision is relatively small, so is the effect on the number of errors. A higher decision ratio than 1.03 had a greater influence on the number of errors, but the recognition rate for both fonts fell beneath 99%.

It is felt that to maintain a high recognition rate, the decision ratio of second-best to best correlation should be equal to or less than 1.03. In this situation the use of a reject mechanism would offer little improvement in total performance. Thus it was felt necessary to add an additional processing pass to handle the confusion groups, such as [П, V, H].

4.4. ANALYSIS OF 3-PHASE CLASSIFIER

The three-phase scheme studied in this project can be basically described as:

1. preliminary classification with centroid registration
2. refined classification through character re-registration
3. additional heuristics to improve the recognition rate

4.4.1. Preliminary Classification

The correlation formula used in the first phase is

$$c(\underline{X}, \underline{M}) = \frac{1}{N} \sum_{i=1}^N (x_i - m_i)^2 \quad (1)$$

where $\underline{X} = \langle x_1 x_2 \dots x_N \rangle$ is the grey-level normalized input character,

$\underline{M} = \langle m_1 m_2 \dots m_N \rangle$ is the grey-level normalized mask of dimension N.

Let t_1 be the time required to compute $c(\underline{X}, \underline{M})$ by a particular processor. Then obviously the total execution time to classify the input \underline{X} in the first phase is

Table 4-5 Font C6 Error Trade-Off Analysis

Ratio of 1st to 2nd Correlation	No. of Errors	No. of Rejects	Error Rate	Recognition Rate
Forced(1.00)	18	0	0.58%	99.42%
1.03	15	10	0.49	99.16%
1.06	10	28	0.32	98.79%
1.09	6	41	0.19	98.42%
1.12	4	78	0.13	97.29%
1.15	0	131	0	95.59%
1.18	0	179	0	94.02%
1.21	0	229	0	92.51%

Table 4-6 Font L8 Error Trade-Off Analysis

Ratio of 1st to 2nd Correlation	No. of Errors	No. of Rejects	Recognition Rate
Forced(1.00)	16	0	99.06%
1.03	16	1	99.02%
1.06	15	3	98.95%
1.09	12	7	98.91%
1.12	9	14	98.75%
1.15	7	18	98.64%
1.18	4	27	98.57%
1.21	2	38	97.87%

recognition
rate

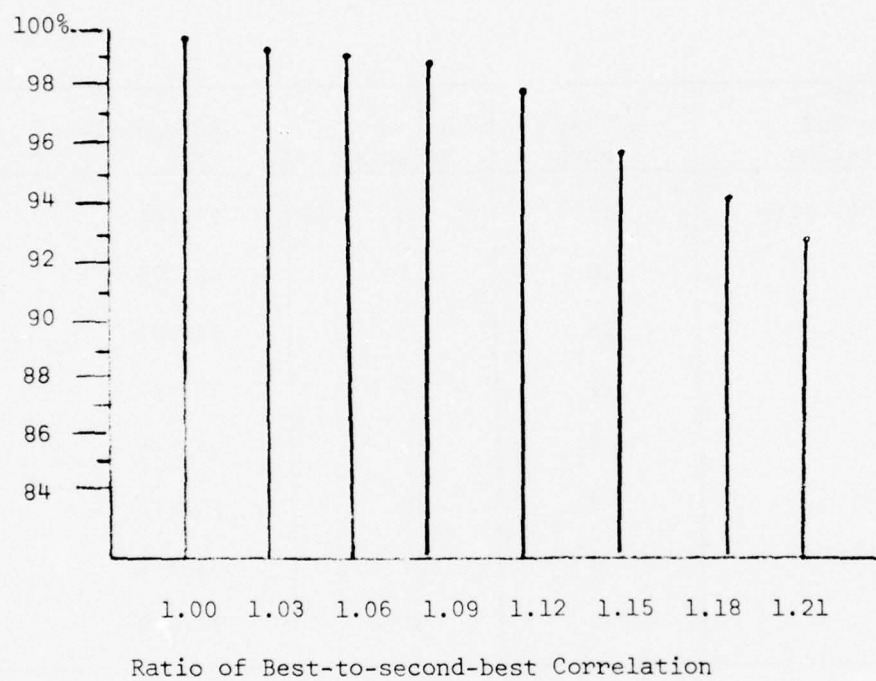


Figure 4-4 Font C6 Error Trade-Off Analysis

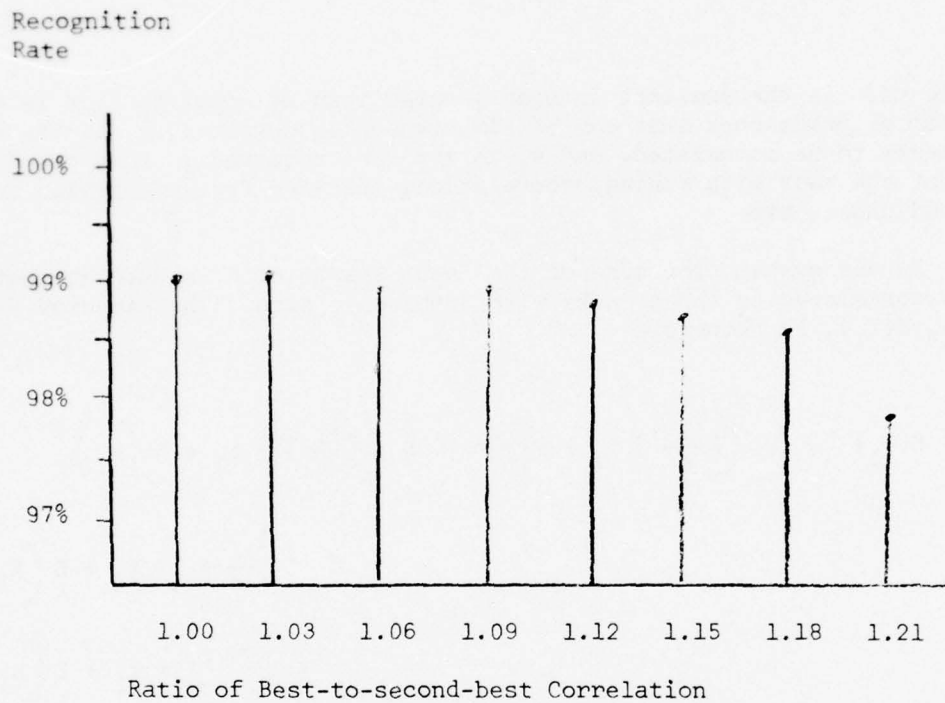


Figure 4-5 Font L8 Error Trade-Off Analysis

$$T_1 = \left[\frac{N_1}{K} \right] t_1 + R_1 \quad (2)$$

where $[X]$ is the smallest integer greater than or equal to X , K is the number of processors that can be simultaneously operated, N_1 is the number of masks to be correlated, and R_1 is the time required to load the data, select the mask with minimal correlation, transfer the information to the second phase, etc.

In our system, the size of the input character X is used so that X is only correlated to those masks with comparable size. The expected execution time, $E(T_1)$, is therefore

$$\begin{aligned} E(T_1) &= E \left\{ \left[\frac{N_1}{K} \right] t_1 + R_1 \right\} = E \left\{ \left[\frac{N_1}{K} \right] \right\} t_1 + E \{ R_1 \} \\ &\leq E \left\{ \frac{N_1}{K} + 1 \right\} t_1 + E \{ R_1 \} \\ &\leq E \left\{ \frac{N_1}{K} \right\} t_1 + t_1 + E \{ R_1 \} \quad (1) \end{aligned}$$

Since N_1 depends on the input character size, we may write

$$E \left\{ \frac{N_1}{K} \right\} = \sum_{i=1}^L P_i E \left\{ |S_i| \right\} \quad (2)$$

where L is the number of alphabets, P_i is the probability of occurrence of the character i in the text, and S_i is the subset of characters with a similar size to that of character i .

Such simple prescreening greatly improves the efficiency of the first phase. Our data show that S_i never exceeds 18 in C8 font with 85% <

$\frac{\text{size}(X)}{\text{size}(M)} < 115\%$. Therefore, for $K=1$, the expected execution time with size selection reduces to approximately one-third of that without (see Table 4-7 for details).

4.4.2. Refined Classification Through Character Re-registration

The same correlation formula (1) is used in the second phase. For r re-registrations and K_2 processors simultaneously operating, the expected execution time will be

$$\begin{aligned} E(T_2) &= E \left\{ \left\lceil \frac{\sqrt{2} \cdot r}{K_2} \right\rceil \right\} t_1 + E \{ R_2 \} \\ &\leq E \left\{ \sqrt{2} \right\} \frac{r}{K_2} t_1 + t_1 + E \{ R_2 \} \end{aligned} \quad (4)$$

where $\sqrt{2}$ is the number of masks required for re-registration.

Again, $E \left\{ \sqrt{2} \right\} = \sum_{i=1}^L P_i' E \left\{ |S_i'| \right\}$ where P_i' is the probability that character i may occur in the second phase and S_i' is the set of characters that cannot be resolved from characters in the first phase. Therefore, $P_i' = P_i \cdot P_i^{12}$ where P_i^{12} is the probability of character i passing to second phase. P_i^{12} depends on the recognition technique, obviously. Table 4-7 also summarizes an estimation of P_i^{12} 's C8 font by using our algorithm. Notice because of the sample size, P_i , P_i' , etc. in Table 4-7 are the frequency of occurrence of character i , expressed as a fraction of unity, as opposed to the probability per se.

Table 4-7 Expected No. of Correlations and Frequency of Occurrence for Cyrillic Data

i	id	A		B		C		D		E		F		G		H		#Pairs ⁵
		$P_i(\%)^2$	E	S_i	³	$P_i^{12}(\%)^4$	$P_i^1(\%)^4$	E	S_i^1	⁴	$P_i^{23}(\%)^4$	$P_i^2(\%)^4$	E	S_i^2	⁴	E	S_i^2	
1	а	7.30		17.43		32.29	2.36	4.23			3.23	0.08		3.00				3
2	б	1.57		5.17		0.00	0.00	-			0.00	0.00		-				-
3	в	4.61		17.33		53.13	2.45	4.94			14.71	0.36		2.00				1
4	г	1.33		13.54		62.50	0.83	3.13			0.00	0.00		-				-
5	д	2.75		5.94		6.45	0.18	2.00			0.00	0.00		-				-
6	е	9.66		13.19		56.59	5.47	3.55			2.74	0.15		2.00				-
7	1																	
8	ж	0.81		6.00		0.00	0.00	-			0.00	0.00		-				-
9	з	1.68		10.28		16.67	0.28	2.00			0.00	0.00		-				-
10	и	8.88		11.41		94.85	8.42	4.94			95.35	8.03		3.03				3
11	й	0.93		5.11		5.56	0.05	2.00			0.00	0.00		-				-
12	к	3.54		13.31		56.86	2.01	5.90			0.00	0.00		-				-
13	л	4.55		13.42		63.51	2.89	7.51			2.13	0.06		5.00				10
14	м	3.25		9.13		9.26	0.30	5.60			0.00	0.00		-				-
15	н	6.67		11.95		95.00	6.34	3.82			37.89	2.40		2.94				3
16	о	10.47		17.62		6.92	0.72	4.91			0.00	0.00		-				-
17	п	2.40		11.23		94.87	2.28	4.60			87.18	1.98		2.97				3
18	р	5.37		13.35		83.53	4.49	3.24			0.00	0.00		-				-
19	т	6.64		15.67		56.25	3.74	2.41			3.17	0.12		2.00				1
20	у	2.53		4.88		0.00	0.00	-			0.00	0.00		-				-
21	ф	0.31		2.00		0.00	0.00	-			0.00	0.00		-				-
22	х	1.03		13.31		0.00	0.00	-			0.00	0.00		-				-
23	ц	0.31		5.00		50.00	0.16	2.25			0.00	0.00		-				-
24	ч	1.33		12.92		36.00	0.48	6.89			0.00	0.00		-				-
25	ш	0.81		2.33		16.67	0.14	2.00			0.00	0.00		-				-
26	щ	0.41		6.67		0.00	0.00	-			0.00	0.00		-				-
27	ъ	2.45		4.63		16.67	0.41	2.80			0.00	0.00		-				-
28	ы	0.37		9.50		70.00	0.26	4.86			0.00	0.00		-				-
29	э	0.49		4.46		0.00	0.00	-			0.00	0.00		-				-
30	я	1.07		11.81		36.11	0.39	3.69			0.00	0.00		-				-
31	0	5.27		6.04		0.00	0.00	-			0.00	0.00		-				-
32	Б	1.19		17.81		80.95	0.96	5.47			5.88	0.06		2.00				1
33	б	0.01		11.00		0.00	0.00	-			0.00	0.00		-				-

1. Deleted
2. Normalized by the total percentage of small letters (98.82%) in the population
3. Size criterion: $0.85 \leq (\text{character size})/(\text{mask size}) \leq 1.15$

Table 4-7 (Continued)

- 4. On 1506 C8 font data set
- 5. Number of pairs when weighted correlation is involved
- A: Frequency of occurrence of the character in the population
- B: Number of masks used for correlation in Phase I
- C: Estimate percentage of characters passed to Phase II
- D: Estimated frequency of occurrence of the characters passed to Phase II
- E: Number of masks used for correlation in Phase II
- F: Estimated percentage of characters passed to Phase III
- G: Estimated frequency of occurrence of the characters passed to Phase III
- H: Number of masks used for correlation in Phase III

Both P_i^{12} and $E\{|S_i'|\}$ also depend on the criteria used to pass the characters to the second phase. In our study, a real number t_{12} was chosen so that the mask \underline{M}' will be passed to the second phase along with \underline{M} if

$$\frac{c(\underline{X}, \underline{M}')}{c(\underline{X}, \underline{M})} < t_{12}$$

where $c(\underline{X}, \underline{M})$ is the minimum correlation between \underline{X} 's and \underline{M} 's obtained in the first phase. It is obvious then that as t_{12} increases, P_i^{12} and $E\{|S_i'|\}$ increase and their $E\{T_2\}$ increase.

4.4.3. Additional Heuristics To Improve The Recognition Rate

Let t_3 be the execution time used for a particular heuristic in order to improve the recognition rate in the third phase (or the fourth, the fifth, etc.). By the same argument, we may write the expected execution time as

$$\begin{aligned} E\{T_3\} &= E\left\{\left[\frac{P_3}{K_3}\right]\right\} t_3 + E\{R_3\} \\ &\leq E\left\{\sqrt[3]{\frac{P_3}{K_3}}\right\} t_3 + t_3 + E\{R_3\} \end{aligned} \quad (5)$$

$E\{R_3\}$, again, is a function of the criteria used to pass the character to the third phase from the second phase.

The pairwise weighted correlation used in the third phase requires, for each mask pair M, M' :

- a. $N \mu_A$ to calculate $W: = |m_i - m'_i|$
- b. $(2N-1) \mu_A + (2N+1) \mu_M$ to calculate $\sum \omega_i (x_i - m_i)^2 / N$
- c. $(2N-1) \mu_A + (2N+1) \mu_M$ to calculate $\sum \omega_i (x_i - m'_i)^2 / N$
- d. 1 comparison to choose the smaller of $c(X, M)$ and $c(X, M')$.

Therefore, $t_3 = (5N-2) \mu_A + (4N+2) \mu_M$ is required for each pair of masks. Notice, for 3 masks input to the pairwise weighted correlation scheme, there are $3(3-1)/2$ pairs to be compared.

When N is large (~ 2500 in our experiments), the time for R_1 becomes negligible and $E\{T_1\} \approx E\left\{ \begin{smallmatrix} 1 \\ 1 \end{smallmatrix} t_1 \right\}$ for $K=1$.

Similarly, $E\{T_2\} \approx E\left\{ \begin{smallmatrix} 1 \\ 2 \end{smallmatrix} \right\} rt_1$ in the second phase and $E\{T_3\} \approx E\left\{ \begin{smallmatrix} 1 \\ 3 \end{smallmatrix} \right\} t_3$ in the third phase.

Estimated values of P_i^{12} , $E\left\{ \begin{smallmatrix} 1 \\ 1 \end{smallmatrix} |S_i| \right\}$, etc., of the Cyrillic characters of font C8 are given in Table 4-7 along with P_i given in [3]. The reliability of these values has not been evaluated because of time limitation. From these figures, one may calculate the $E\left\{ \begin{smallmatrix} 1 \\ 1 \end{smallmatrix} \right\}$, $E\left\{ \begin{smallmatrix} 1 \\ 2 \end{smallmatrix} \right\}$, $E\left\{ \begin{smallmatrix} 1 \\ 3 \end{smallmatrix} \right\}$ in sequential simulation as

Two heuristics were used in the study. One is a general weighted correlation (Section 3-6) and the other is a feature-looking technique specially designed to resolve the set [И, И, П] in Cyrillic alphabets. Their time complexity will be analyzed in the following subsection.

4.4.4. Time Complexity In Sequential Simulation

In order to discuss the cost-performance trade-off of a practical OCR system, one must analyze the time required for character recognition. Such analysis is very difficult, if indeed possible, unless the system has been determined in detail, especially when parallel processing is involved. The following sequential simulation analysis can only give a first order approximation.

In the first two phases, it requires N subtractions, N multiplications, $N-1$ additions, and one division in order to obtain

$$c(\underline{X}, \underline{M}) = \frac{\sum_{i=1}^N (x_i - m_i)^2}{N}$$

Therefore,

$$t_1 = (2N-1) \mu_A + (N+1) \mu_M$$

where μ_A and μ_M are the time required to perform one addition and one multiplication, respectively.

$$E\{A_1\} = \sum_{i=1}^{33} P_i E\{|S_i|\} = 12.43$$

$$E\{A_2\} = \sum_{i=1}^{33} P_i' E\{|S_i|\} = 1.97$$

$$E\{A_3\} = \sum_{i=1}^{33} P_i'' E\{|S_i''|\} = 0.39$$

Therefore

$$E\{T_1\} = 12.43 t_1$$

$$E\{T_2\} = 7.87 t_1 \quad \text{for } r=4$$

$$E\{T_3\} = 0.39 t_3$$

If we further assume that 1 multiplication consumes twice as much time as 1 addition, then $t_3 \approx 3.25 t_1$ and $E T_3 = 1.27 t_1$.

According to Table 4-8, the recognition rate for font C8 after phase 1 is 94.82%, after phase 2 is 98.76%, and after phase 3 is 98.89%. Therefore, it seems the performance of the weight correlation in phase 3 is not worth the time cost for the sequential processing.

The feature-looking technique is used when the input character has been determined as belonging to [H, И, П]. It requires

Table 4-8 Number of Characters Reaching each Stage of Classification

Font	No. of Chars. in 1st Pass	No. in 2nd Pass, %		No. in 3rd Pass, %	
		#	%	#	%
C1	1654	717	44%	166	10%
C2	2038	596	29%	*	*
C4	1972	144	7%	*	*
C6	3085	1802	58%	*	*
C8	1506	706	46%	205	14%
L1	1599	608	38%	48	3%
L2	1128	420	37%	76	7%
L4	1491	608	41%	54	4%
L5	1396	600	43%	60	4%
L8	1742	877	51%	179	11%
Total**	10516	4536	43%	788	7%

* Information unavailable at processing time

** Total only includes the 7 fonts for which all information is available

- a. $\frac{1}{4} \cdot N \mu_A$ to get the vertically projected values in the central part of the input.
- b. $\frac{1}{2} \sqrt{N}$ comparisons to find the location of the peak if there is any (here we simply assume that the dimension of the input is \sqrt{N} by \sqrt{N}).
- c. 2 comparisons for decision making.

Assuming 1 comparison requires 1 add-time, we have

$$t_3 = \left(\frac{1}{4} N + \frac{1}{2} \sqrt{N} + 2 \right) \mu_A \approx \frac{1}{8} t_1$$

Since

$$P_H'' + P_M'' + P_\Pi'' = 14.08\%, \text{ the estimated execution time}$$

$$E \{T_3\} = \frac{1}{8} * 0.1408 t_1 = 0.018 t_1$$

According to our study, 84% of the error is due to the indistinctness among Ψ , Π and Π . For the C8 font, this means the error rate can be reduced to 0.18%. That is, by spending 0.09% more execution time ($0.018 / (12.43 + 7.87)$), one may get an 0.95% increase in recognition rate. It is therefore recommendable for sequential processing.

4.4.5. Performance Study

The Latin font results are analogous to the Cyrillic results. The second phase resolved 65 of the 88 total first phase errors. The third phase offered no improvement over the second phase.

Table 4-8 provides information about the number and percentage of characters that reached the second and third phases. The criterion for determining if a second and third phase are needed is the following: the ratio of the correlation of a given character-mask pair to the lowest correlation of all the character-mask pairs is computed for all character-mask pairs. All the masks which have a ratio less than 2.5 are subjected to the second phase of correlation. The correlations are again computed as are the correlation/lowest correlation ratios. All the masks which have a ratio of less than 1.4 passed on to the third phase of correlation where the weighted difference masks are employed. The sample is then classified as the character which corresponds to the lowest character-mask correlation. If, at the end of any phase of correlation, only one character-mask pair meets the ratio criterion, then a classification is made and the process is terminated. An example of this classification method is illustrated below; the "c" is correctly labelled a "c".

Table 4-9 presents the recognition after each phase of classification. Using only center of mass registration resulted in an average Cyrillic recognition rate of 97.27% with 2 fonts, C1, 92.62% and C8, 94.82%, doing poorly. After the additional +1 pixel shifting, the average rate rose to 99.43% with the 2 troublesome fonts, C1 and C8, significantly improved. Because the mask weighting algorithm was not fully developed at the time that fonts C2, C4, and C8 were processed, the third phase results are not available for those fonts. However, of the other two fonts, the third phase resolved only 2 of the 19 C8 errors and none of the 13 C1 errors.

Table 4-8 reveals that, on the average, 43% of the samples of a font are passed to the second phase and 7% of all the samples were forced passed to the phase. The recognition rates rose significantly in the C1, C8 and L8 fonts with the use of the second phase. The overall number of Cyrillic errors dropped from 280 to 59 with the use of the second phase. Using the time analysis results obtained, this can be expressed as 2.16% (99.43%-97.27%) decrease in errors for a 63% ($7.87 t_1 / 12.43 t_1$) increase in computer time. The decrease in errors using the weighted correlation in the third phase does not justify the increase in computer time.

Character	Mask	1st Pass		Ratio to		2nd Pass		Ratio to		3rd Pass	
		Correlation		Lowest		Correlation		Lowest		Correlation	
c	a	8000		2.00		6000		2.00			
	c	4000		1.00		3000		1.00		3000	
	e	5000		1.25		5000		1.67			
	o	6000		1.50		4000		1.33		4200	
	x	13000		3.25							

Sample of Classification Method

Table 4-9 Recognition Results for each Stage of Classification

Font	No. Char.	Errors After Registration, Recognition Rate		Errors After Shifting, Recognition Rate		Errors After Mask Weighting, Recognition Rate	
		#	%	#	%	#	%
C1	1654	122	92.62%	13	99.25%	13	99.25%
C2	2038	15	99.26%	9	99.53%	*	*
C4	1972	3	99.85%	0	100 %	*	*
C6	3085	62	97.99%	18	99.39%	*	*
C8	1506	78	94.82%	19	98.76%	17	98.89%
Cyrillic Subtotal	10255	280	97.27%	59	99.43%		
L1	1599	9	99.44%	1	99.95%	1	99.95%
L2	1128	7	99.38%	3	99.84%	3	99.84%
L4	1491	8	99.46%	2	99.91%	2	99.91%
L5	1396	12	99.14%	1	99.92%	1	99.92%
L8	1742	52	97.01%	16	99.06%	16	99.06%
Latin Subtotal	7356	88	98.80%	23	99.74%	23	99.74%
Overall Total	17611	368	97.91%	82	99.53		

* Information unavailable at processing time

SECTION 5

IMPROVEMENTS TO THE RECOGNITION SCHEME

Figure 5-1 illustrates a problem that occurred in the character isolation section of the OCR system. To allow for the possibility of overlap of neighboring characters, the extraction program DOCR searches every 90 elements for the minimum number of non-zero pixels in the line of text vertical histogram. The "t" and "h" in line 2 of Figure 5-1 overlap while there is a zero pixel count after the "h". Since the combined width of the "t" and "h" characters is less than the maximum allowed width, these two were isolated as one character, "th".

Another isolation problem is illustrated in Figure 5-2. This takes place when the non-zero pixel count minimum occurs before the end of the character encountered. This can be a result of a character width greater than 90 pixels; for example, some Cyrillic capitals, or when there occurs a printing flaw and there exists a break in the character itself.

One possible solution to these isolation difficulties is to have the extraction and classification subsystems interact with one another. When the classifier rejects a character or consecutive characters, these characters should be subjected to another level of extraction. The width of the "character" after the first isolation would reveal the existence of two or more overlapping characters. These overlapping characters can be isolated by searching for a minimum within the extracted "character". This would, however, require storage of the raw scanned data for all characters not completely processed, or the ability to rescan from the printed page.

When successive characters are rejected, this could also be an indication of "over-isolated" characters. The consecutive rejects could be concatenated together to form a new "character" and be submitted for re-classification.

As noted in Section 4.1., almost 85% (48/57) of the errors in the Cyrillic fonts occurred among the М, Н, and П characters. The following discussion will provide insight into the cause of classification errors and an alternative method of classification for these three characters.

Figures 5-3 through 5-5 present these three masks for the C2 font. For illustrative purposes, the intensity values for each pixel have been quantized into two levels, symbolized by a blank or "0". Figure 5-6 is a sample "H" from font C2 also quantized into two levels, but symbolized by a blank or "x". Figure 5-7 shows this sample overlayed with the H mask. Figure 5-8 shows the difference between the mask and the character (quantized into several levels). Figures 5-9 and 5-10 show the H character overlayed with the П mask and the difference between the two, respectively.

Line 1

id that in bot

IMAGE

2

when the foc

3

riktion to non

Row 2

Extracted
Characters

W h e n t h e f o

Figure 5-1 Example of Character Overlap.
Tick Marks Denote Starting Location of Each Character

3 T C

Figure 5-2 Example of Split Character

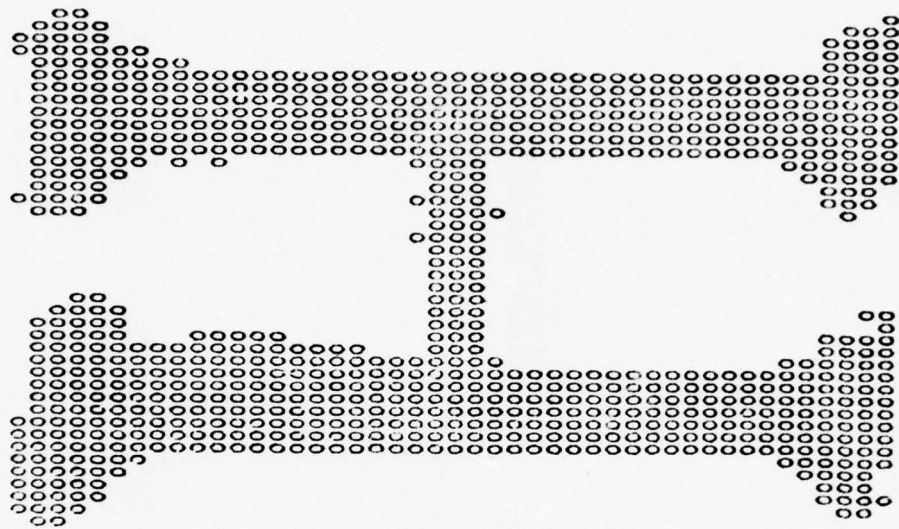


Figure 5-4 H Mask, Font C2

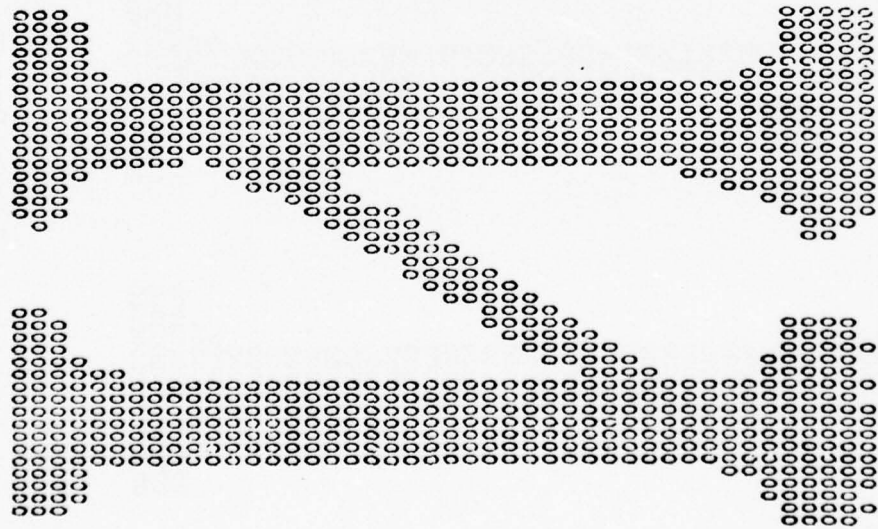


Figure 5-3 M Mask, Font C2

A large, stylized letter 'A' composed of 'x' characters. The letter is formed by a grid of 'x's, with a small 'x' at the top center and a small 'x' at the bottom center. The 'A' is symmetrical and has a solid, blocky appearance.

Figure 5-5 II Mask, Font C2

Figure 5-6 H Sample

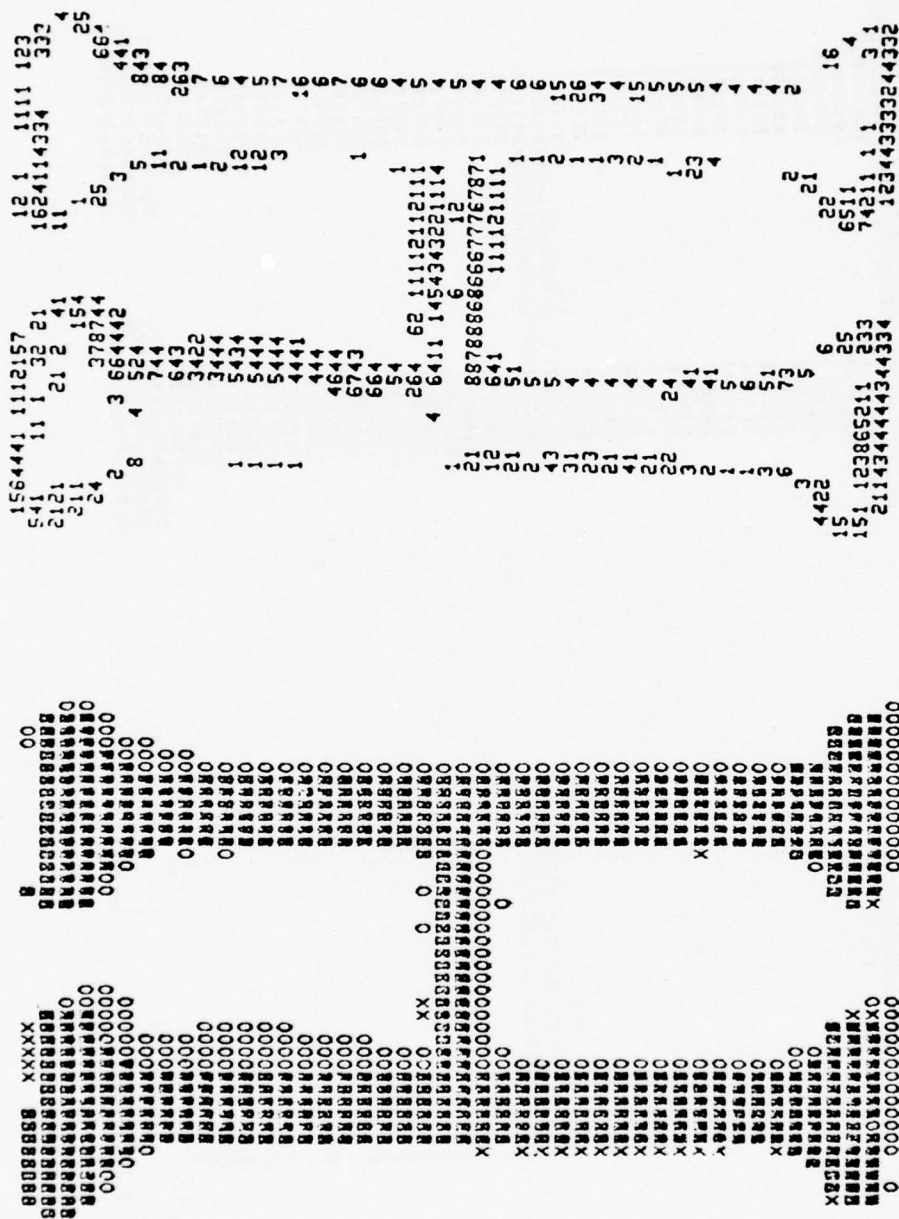


Figure 5-7 H Mask, H Sample Overlay

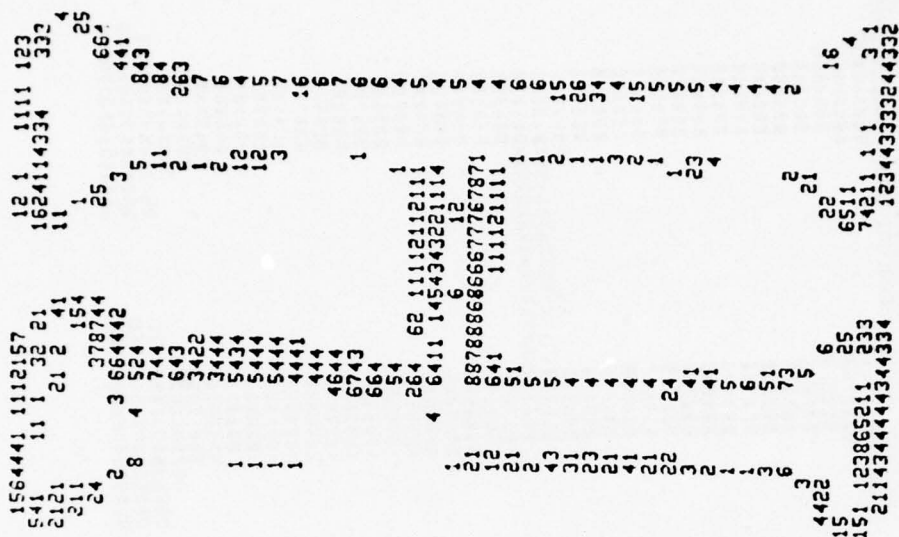


Figure 5-8 H Mask, H Sample Overlay Difference

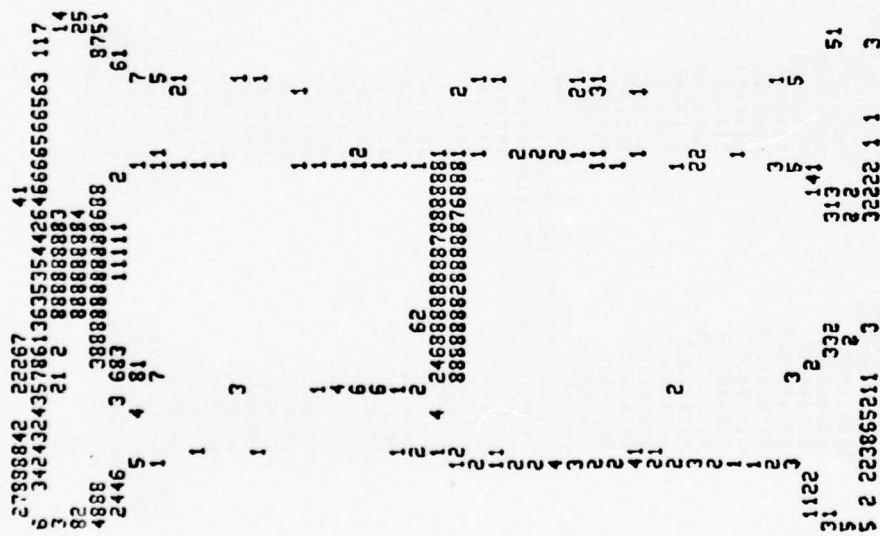


Figure 5-9 II Mask, H Sample Overlay
Difference

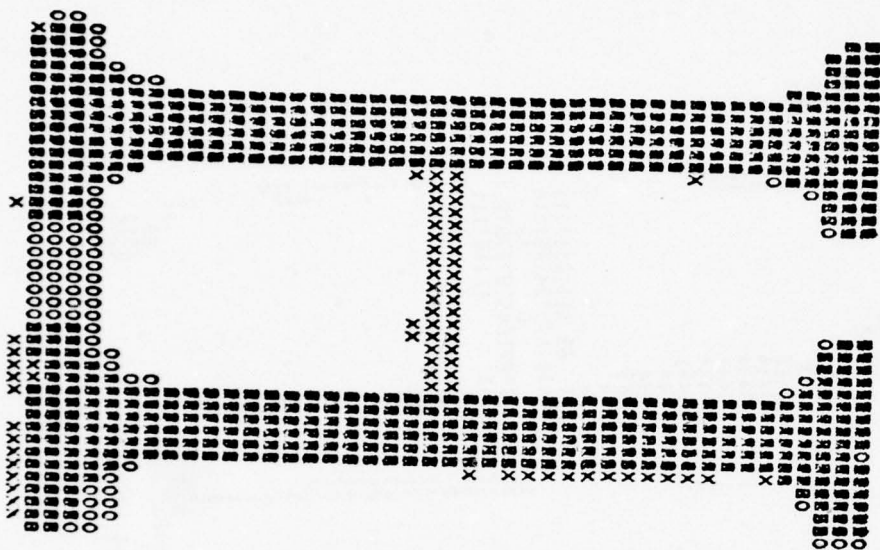


Figure 5-10 II Mask, H Sample Overlay

Comparison of Figures 5-8 and 5-10 shows that the least amount of overlap occurs in Figure 5-8, the H character and the H mask, although the difference in the amount of overlap between the two comparisons is small. This is reflected in the difference between the H/H correlation of 5968 and the H/Π correlation of 7012. This character was correctly classified as H.

Figures 5-11 through 5-15 present another H sample, as above, which was incorrectly classified as a Π. The H sample is slightly thinner in the vertical strokes than the previous sample and slightly shorter in height than the H mask. There also exists a chopped or mutilated serif. The decision between H and Π is very close with a H/Π correlation of 9484 and a H/H correlation of 9604, a difference of only 1.3%.

The regeneration of the H mask by selection of a higher threshold to reduce the "spread" of the mask could possibly create a correct decision in the second example cited above. This technique of higher thresholding of certain masks generally decreased the error rate to an acceptable level. However, there are two problems associated with this:

1. The iterative process of selecting the best combination of differently thresholded masks is operator-dependent and requires many runs of data to determine what is the best combination, and
2. Adjusting for instances of a specific error type often creates other errors when applying a given combination of masks to more data.

Since 85% of the Cyrillic errors occurred in the H, H, Π subset, an alternate level of logic was introduced for these three characters to reduce the overall error rate. After a sample has been identified by the classification system as a member of this subset, it is subjected to an area-sensitive logic that examines the main area of difference between these characters. A vertical or column histogram is made and the area between the two vertical segments of the character extracted. See Figures 5-16 through 5-18. A horizontal histogram is then performed over the extracted area. This area is enclosed by the dotted line in Figures 5-16 through 5-18. The resulting distribution readily identifies the character as a H, H, or Π.

The C8 font was selected as the test set for this algorithm because all of the 17 errors in this font were misclassified H's and Π's. The error rate was reduced to 0%.

As a result of this investigation, it is suggested that this alternate logic for the three character subset be incorporated into the basic correlation system. This also suggests that other confusion groups may be resolved with special logic.

Figure 5-11 H Sample

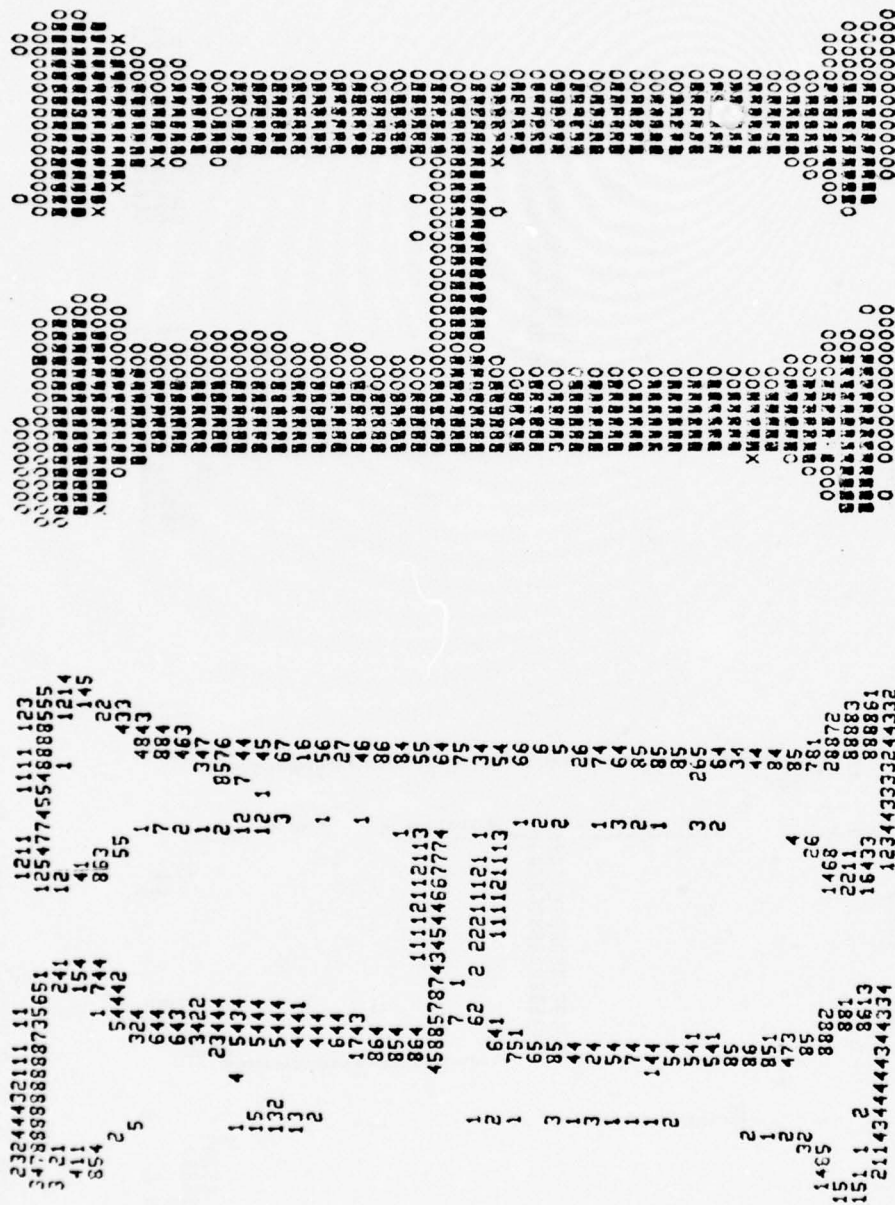


Figure 5-12 H Mask, H Sample Overlay

Figure 5-13 H Mask, H Sample Overlay Difference

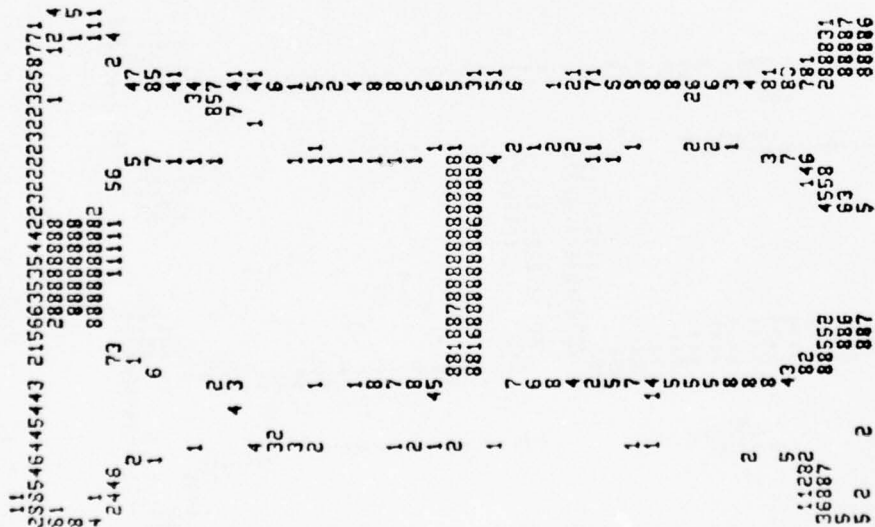


Figure 5-14 H Mask, H Sample Overlay
Difference

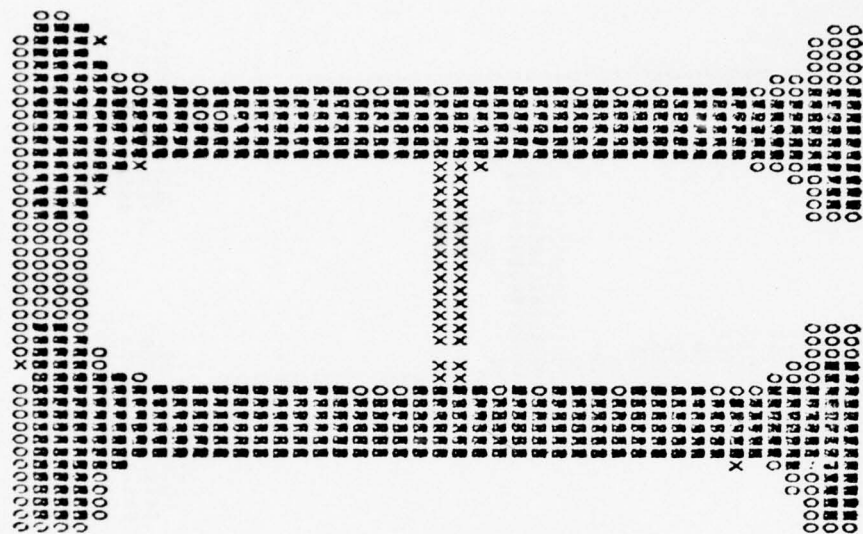


Figure 5-15 H Mask, H Sample Overlay

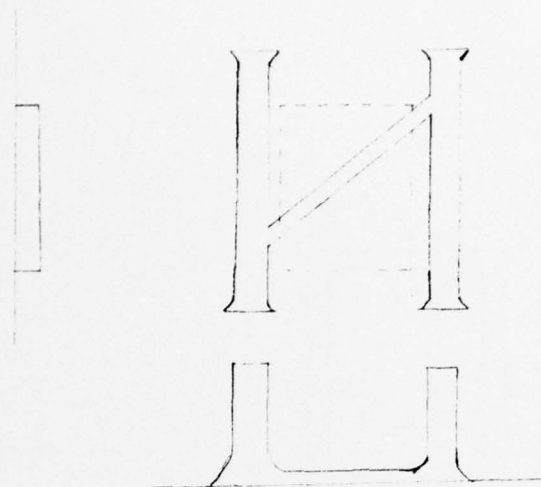


Figure 5-16 Row and Column Histograms of M Character



Figure 5-17 Row and Column Histograms of E Character

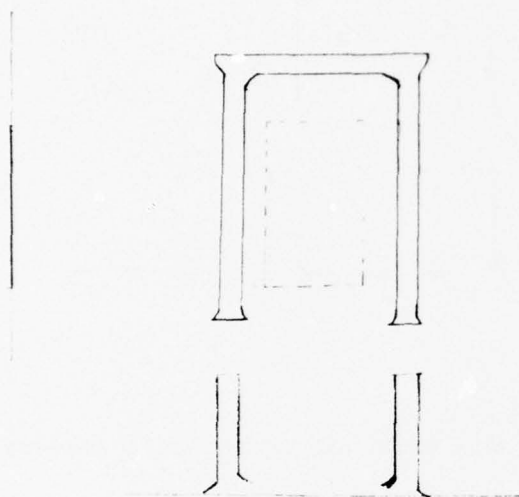


Figure 5-18 Row and Column Histograms of Π Character

The price of using special logic is, at least initially, the requirement for manual design thereof. Thus the simple mask design scheme originally envisioned for the correlation approach would no longer be adequate to achieve the highest level of performance possible. The inability to automatically train on-line has to be evaluated in the context of the intended application(s). It is expected that if the relative frequency of new font design is low then on-line design is not a necessity. In the foreign journal translation case, after the initial designs are done, the new font frequency would be determined by the publishers of new and/or old journals and the rate these are acquired by the Government. The publishers would have to automate their font design for printing to make it worthwhile to automate our font design for reading. In the meantime, manual design of new fonts for OCR classifier logic could be provided quickly and efficiently by skilled government and/or contractor staff who are familiar with the system specifications, and who understand OCR design principles.

SECTION 6

HARDWARE CONSIDERATIONS

6.1. BACKGROUND

The concept of a multi-font machine is not new. Several approaches have been suggested and implemented. The OCR unit built for Social Security provides separate firmware recognition logic for all expected fonts[4]. A more efficient, but probably less accurate, approach would use a small number of sophisticated recognition logics which can generalize over several fonts each. Another approach allows the OCR to be reconfigured at run-time by reading into a control memory the recognition logic parameters appropriate for the current run. An adaptive approach which requires each character to be identified correctly by a human trainer has been promoted as providing a semi-automatic means to extend the basic repertoire of an OCR[5].

The firmware approach is expensive and still is limited to a fixed repertoire of fonts. The lead-time to add a new font is relatively long. It has been proposed to use hand-printing recognition logic to provide the generalization capability over many machine print fonts because its recognition logic must perform well for many styles of printing and generally poorer character print quality. However, this added sophistication was not needed for installations reading only a few different OCR fonts and therefore not commercially developed. Furthermore, this logic would not take advantage of the a priori knowledge of the font size and shape specifications. Machine print differs from hand-print basically in its geometric repeatability. The machine reconfiguration-at-run-time approach economizes on hardware costs, by requiring a smaller number of more standard logic units, each having memory loaded under software job control. This technique is also limited to those fonts available at the moment, but new fonts can be added using off-line software recognition logic design routines. If quicker adaptation to new fonts is required, on-line design can be obtained and implemented in software using adaptive training procedures. However, the throughput (processing speed) performance resulting is slow, because of software logic implementation, and the accuracy is highly sensitive to errors introduced in training by insufficiently skilled operators. In order to completely resolve certain character-pair confusions, it is probably necessary to design special logic that can be applied to just those cases. This suggests the need for off-line design by skilled technical personnel. The design techniques would use a simulation of the basic hardware classifier. The resulting data would then be used by the on-line hardware to reconfigure its operation appropriately.

Correlation search for the best match is performed on digital images sampled in rows and columns by shifting the unknown character image left

and right, and/or up and down, about the centroid of each mask, and choosing the lowest variance (highest correlation). The present variance function (implemented in the DIMES system on the HIS 635 and in P1/I on the HIS 6180) also normalizes the effects of different average grey-levels and contrasts (intraimage variances). However, there are no explicit normalizations for rotation, skew, and non-isotropic scale changes. The effects of these variable were previously investigated.

Successful character recognition depends not only on powerful classification logic, but also on character detection, location, isolation, and registration techniques. Once the next character to be read is located and isolated from its neighbors and from extraneous markings, the recognition routine must accurately register the character and each of the stored masks. Detection, location, and isolation are usually performed by logic separate from the classification logic, especially when automatic tracking of lines of text is required. However, correlation values can also be used to obtain a measure of registration to obtain the best fit when nearly registered. Thus, it is possible to integrate the classification logic with a control algorithm for character registration. As described above in Section 3, such integration was included in the tested design.

Nearly all OCR's depend on constant pitch and fixed formatting (as opposed to free form) of characters to be read. In addition, it is usually known what the font is at each field (space allocated for a single logical data item). In this case, the need to handle typeset characters, variable spacing, freeform formatting, and unpredictable font type and size must be accommodated. An additional complication which occurs, particularly in technical journals, is the presence of graphics. Line drawings and illustrations can occur at unpredictable locations on a page of text as well as cover entire pages. Manual cut-paste-and-scan operations are the common way of obtaining these data at present. The control algorithm for detecting and locating the text, data, and graphics must perform well in spite of all of these peculiarities.

The work, under contract F30602-75-C-0269, was conducted in the context of an eventual hardware implementation. That is, the knowledge of current and announced computational components and techniques has strongly influenced the direction and form of the design evaluation software.

6.2. THE PROPOSED TRIAL HARDWARE FOR CLASSIFIER LOGIC

Hardware construction for the proposed OCR system could be accomplished in three phases. The first phase would comprise the development of a limited character processor subsystem which implements the classifier logic and interfaces. Phase 2 will put this subsystem on line with the specified host computer. A third phase could expand the system to meet the total performance specifications.

A PDP-11/45 such as one of several now located at PAR, could simulate the host computer for Phase 1. The 11/45 itself is incapable of supporting the desired processing speeds necessary for the functions of inputting scanned text, and detecting, locating, and isolating characters with the desired resolution; but that is not deemed necessary for the Phase 1 effort. Its primary function will be simply to transmit to the character processor subsystem previously digitized and isolated characters for the purpose of recognition, to receive back the decision or rejection information, and to tabulate the performance statistics.

The character processor subsystem proposed will consist of several microprocessors organized as shown in Figure 6-1. It consists of one microprocessor which serves as character processor input controller; this microprocessor must distribute character information to the other microprocessors. A second microprocessor serves as the character processor output controller and as arbiter of their decisions.

The remaining microprocessors are character processors. They each have access to sufficient memory to store the masks for the characters in the font(s) of interest. The function of the proposed character processor is to determine a measure of the correlation between the mask in its memory and the particular character being processed. A norm monotonically related to Euclidean distance will be used to determine correlation. Also, special logic will be used to discriminate between characters that tend to look alike except for differences that are small in area.

The proposed character processor controller will poll the character processors to determine when a processor is ready to receive a new character or transmit the results of the previous character identification. The input controller must assign a sequence number to each character and direct the loading of this character into the memory of a "ready" character processor.

The output controller must receive the identified characters and output them in proper sequence. Unidentified characters are directed to the error correction routines.

Although the ultimate design of the proposed hardware will include sufficient capacity to process each character in the fonts of interest, Phases 1 and 2 provide for the building of a limited subsystem, sufficient to determine hardware feasibility.

By using only two to four character processors in the limited subsystem the processing speed will be limited, if all masks are used. However, the ability of the system to use different sets of masks as its alphabets

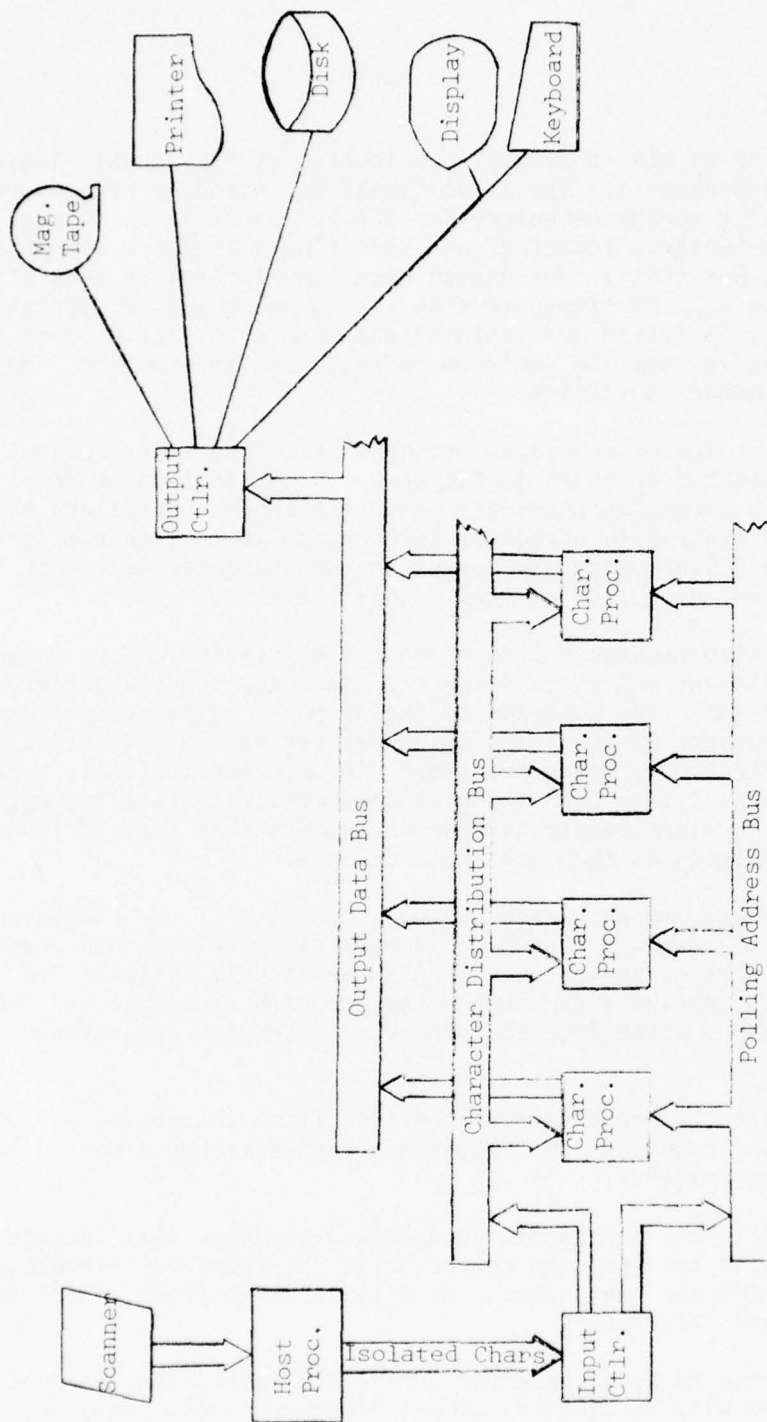


Figure 6-1 Proposed Character Processor Subsystem

enables a wide range of system tests to be completed. The essential considerations in the hardware design will be answered with this amount of circuitry.

The proposed hardware can be built with attention given to appropriate parameters of minimum noise interference and cost with maximum chip density, transmission rates, and processing speeds.

A cross assembler would be purchased for the purpose of translating the assembler language programs for the micro processors into their machine language.

It is felt that the experience gained on the design development, test, and evaluation effort reported herein, combined with general expertise obtained by direct responsibility for several OCR hardware projects, in all phases thereof, is necessary for efficient execution of the suggested companion classifier hardware effort, described below.

6.3. TECHNICAL APPROACH

The hardware-software configuration will automatically classify (when possible) a subset of all characters on digitized images of typeset Latin (English) or Cyrillic (Russian) text, previously scanned from technical journals and other sources. It will also provide a facility for different subsets of those characters though its initially limited scope renders simultaneous capability for all characters impossible.

The software system on the Phase 1 host computer will assume that the automatic detection, location, and isolation of individual characters will have been performed. These isolated characters will be presented to the classification logic to be either recognized or rejected.

The character processor hardware will consist of sufficient hardware to realize a character recognition capability. In addition to such miscellaneous equipments as power supplies and DMA hardware for communication with the host computer (a PDP-11/45), the hardware produced will include the character processor controller and the character recognition logic for two to four character processors. The character processor subsystem will be constructed with printed circuits and/or sockets to permit wire-wrapped connection of the integrated circuits. Backplane wiring will be provided. Intel 3000 series micro-processors will probably be used for the character processor and character processor controller; however, final selection of components will be based on the cost, capability, and availability of devices current at the time the work is undertaken.

Programs to implement the required functions on the microcomputer will be written in the appropriate assembler language. To translate the assembler language to the necessary micro machine language, a cross assembler which runs on the 11/45 will be purchased and modified as necessary.

The software system will provide a convenient means to store the classification results and the images of characters rejected and misclassified.

6.4. PROCESS TIME ESTIMATE

An estimate has been made of the processing time required for character recognition in the character processors. The assumed processor is a bi-polar micro-processor with a total system cycle time of 150ns utilizing 16 bit words. The recognition process consists of three decision levels where correlations between the unknown data character and stored character masks are evaluated. As characters are identified they exit from the recognition process and therefore all characters do not reach Phase 2 or Phase 3 of the process. The estimated processing time is the average time per character taking into account the statistical frequency of occurrence for the characters as well as the number of correlations and process phases required to identify each character.

The timing estimates were made using a sample character set consisting of 32 lower case Cyrillic (Russian) alphabetic characters. The total average processing time per character is 144 milliseconds. This represents a straight line approach without the application of timesaving techniques.

To meet the specified processing rate of no more than 10 seconds per page (2000 characters per page avg.), some time reduction techniques will be employed. Among the more significant techniques to be evaluated is the averaging of adjacent horizontal and vertical pixels for a possible data reduction factor of 4, and the use of parallel character processors with a time reduction factor proportional to the number of processors used.

Assuming full utilization of the two techniques mentioned above, it would require eight character processors to meet the requirement of 10 seconds per page or 5 ms per character for the 32 character test font. If the number of different characters processed at one time is increased, the average time per character will increase. The exact amount of increase relative to a particular expansion of the character set has not been determined at this time.

SECTION 7

SUMMARY AND FUTURE WORK

The project effort resulted in an evaluation of a correlation technique for implementation in a hardware character recognition processor. The source of the five Cyrillic and the five Latin fonts processed experimentally using design simulation software was several Russian technical journals. A total of 17,611 characters (7356 Latin and 10255 Cyrillic) were processed. The correlations were performed on the RADC HIS 6180 computer facility under the MULTICS time-sharing environment.

It was shown that presorting the samples by size will eliminate the inclusion problem; that is, some characters are included in others as parts and obtain high correlation with those parts. Presorting also reduced overall computation time. Registration of sample and mask by center of mass resulted in an overall recognition rate of 97.91%. When the method of shifting masks ± 1 pixel in each direction was added, this rate rose to 99.53%. The weighted mask technique increased this rate to 99.56%.

A time-analysis study showed the 1st and 2nd stages of the classifier to be cost-effective. The 3rd stage of weighted difference masks takes a large proportion of computer time to resolve a small number of errors. In view of this, a new algorithm was devised for implementation in the most error-prone confusion group of Cyrillic characters, И, Н and

П. This procedure was tested in the C8 font and increased the recognition rate to 100%, using 1506 characters.

An error vs. rejection trade-off analysis showed that to reject any proportion of erroneously classified characters, a greater proportion of correctly classified characters must be rejected.

A preliminary study of entropy, average intensity, and average length of black/white segments revealed these statistics to be reasonably consistent over any region of either text or non-text. Whether these statistics are individually sufficient to distinguish various text from non-text requires further investigation.

To increase the effectiveness of the character isolation function, interaction between the extraction and classification procedures is needed. That is, the classifier should relay its results to the extraction system for use in re-isolation of rejected characters.

A hardware throughput study determined that design limitations prohibit the use of an average of 2000 pixels per character. That is, the use of a 40 micron "spot" results in too much image data to economically process 200 characters per second. A simulation of lower resolution by reducing patterns 50% linearly on a 1200 character subset increased the error rate for the

sample from .6% to approximately 1%. It will be necessary to study specific application requirements to determine if the increase in error rate is tolerable. If smaller error rates are required, either more processor units and/or some reduction in speed will be needed.

Presorting the samples by size has shown to lead to a reduction in computation. Another area of investigation which should be examined is the use of conditional probabilities to bias the classification based on the identity of the previous sample. A method for identifying specific fonts needs to be determined; the alternative is a possible six-fold increase in the number of parallel processors. It should be determined whether an operator should enter a font type or just certain characteristics of each batch run, e.g., journal name, which lend themselves to automatic identification. An improved method for creating masks or training the machine when new fonts are encountered needs to be devised. Certain character confusion groups appear common to several fonts. More experimentation is needed to substantiate this possibility. The alternative is that many fonts will require some special logic, possibly designed manually, to resolve the problem cases. Completely automatic design schemes remain a concept, not yet proven in practice, when correlation logic is used.

This study was conducted in the context of an eventual hardware implementation. The idea of parallel microprocessors to compute the necessary correlation is essential. Specialized hardware is needed to improve the throughput. The average forced recognition error rate of 0.5% is encouraging considering the fact that the source material was not meant to be read by OCR equipment. To verify an error rate of, say 0.5%, to a confidence level of 95%, requires the processing of about the number of characters used on this project. However, if better recognition is achieved it will be necessary to process many more characters to be confident that the measured error rate will hold in practice. To use computer simulation to process such a large amount of data, e.g., 10^5 to 10^6 characters, would be uneconomical. It is recommended therefore that the next step be to implement a complete hardware prototype, if it is necessary to verify error rates of 0.1% or less. Before going to the expense of building a complete prototype that will scan the printed page, isolate and classify characters, it should be shown that it is probable that 0.1% error rate can be achieved. This effort suggests that the unaided correlation scheme will not achieve error rates less than 0.5% even at high resolution (40 micron spacing between samples). It also showed that special logic that was specifically designed and added to the basic scheme to resolve special sets of confusion characters, was quite effective in lowering the error rate on the data to which it was applied. It is recommended that this approach be continued and tested on more fonts.

REFERENCES

1. "OCR Software Development", Final Technical Report, RADC-TR-75-232, Sept. 1975, (A018713).
2. IBM Research Report 140-68.
3. "Engineering Analysis And Digital Simulation of the Optical Russian Printhead", Technical Documentary, Report No. RADC-TDR-62-472, Sept. 3, 1962, (401886).
4. "The IBM 1975 Optical Page Reader, Parts I, II, and III" IBM J. Res. Develop., Vol. 12, No. 5, Sept. 1968.
5. "SWAMI", Scan-Data Product Development Literature, PDS14, July 1971.

APPENDIX A
SOFTWARE DOCUMENTATION

(Program descriptions and file structures developed for this project are included with accompanying flowcharts).

Program Name: char_process

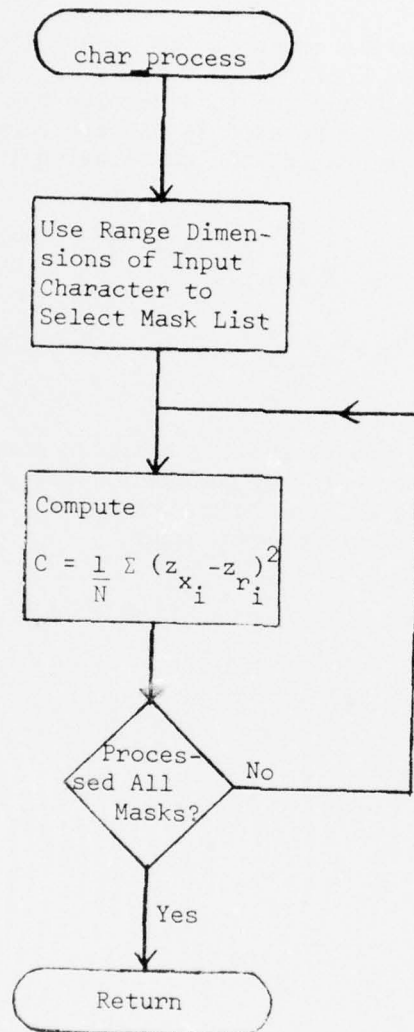
General Program Description:

This subroutine computes the correlation for the first pass of the three-pass classifier. The input character's dimensions are multiplied by 1.14 and .86 to determine the range of dimensions of possible masks to be used in the correlation. When the list of masks has been generated, the correlation is then computed according to

$$C = \frac{1}{N} \sum_{i=1}^N (Z_{x_i} - Z_{p_i})^2$$

$$Z_{x_i} = \frac{x_i - \bar{x}}{\sigma_x}$$

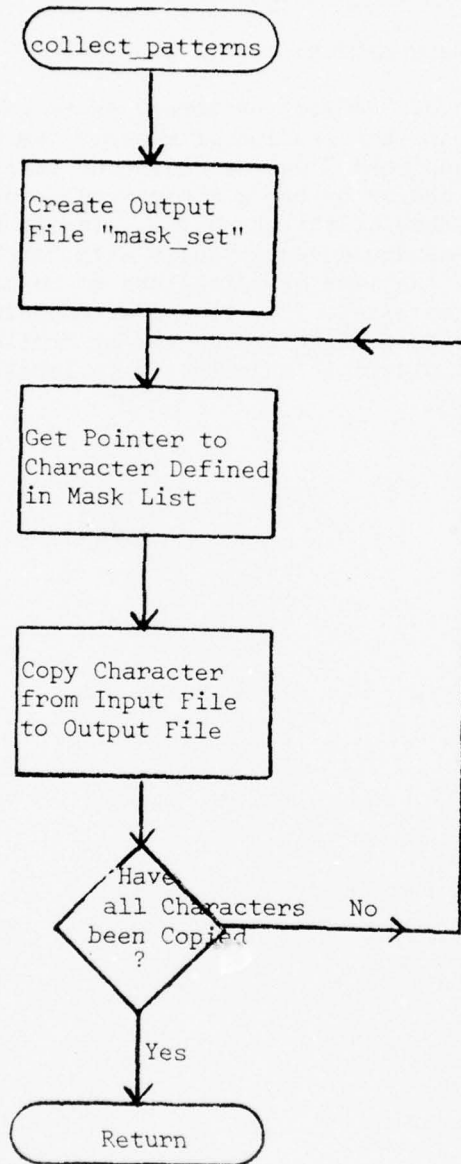
The normalized variable z has been computed by "transformer" prior to execution of char_process. This process is repeated until all mask candidates have been correlated. The program exits by returning control to "correl_main".



Program Name: collect_patterns

General Program Description:

The function of "collect_patterns" is to gather various transformed patterns for use in the creation of masks. The mask patterns are created by executing "transformer" on the character files after they are isolated by "docr". Each character has a three-word header which contains the identification number of the character. In the Latin fonts, the identification numbers are assigned sequentially with "a" having number 1 and "z" having number 26. The list of characters to be used by "collect_patterns" is generated by "mask_select". This list contains a data file name and index number of each pattern to be used by "collect_patterns". The characters are collected and placed in an output file labelled "mask_set" for use by "mask_generator".

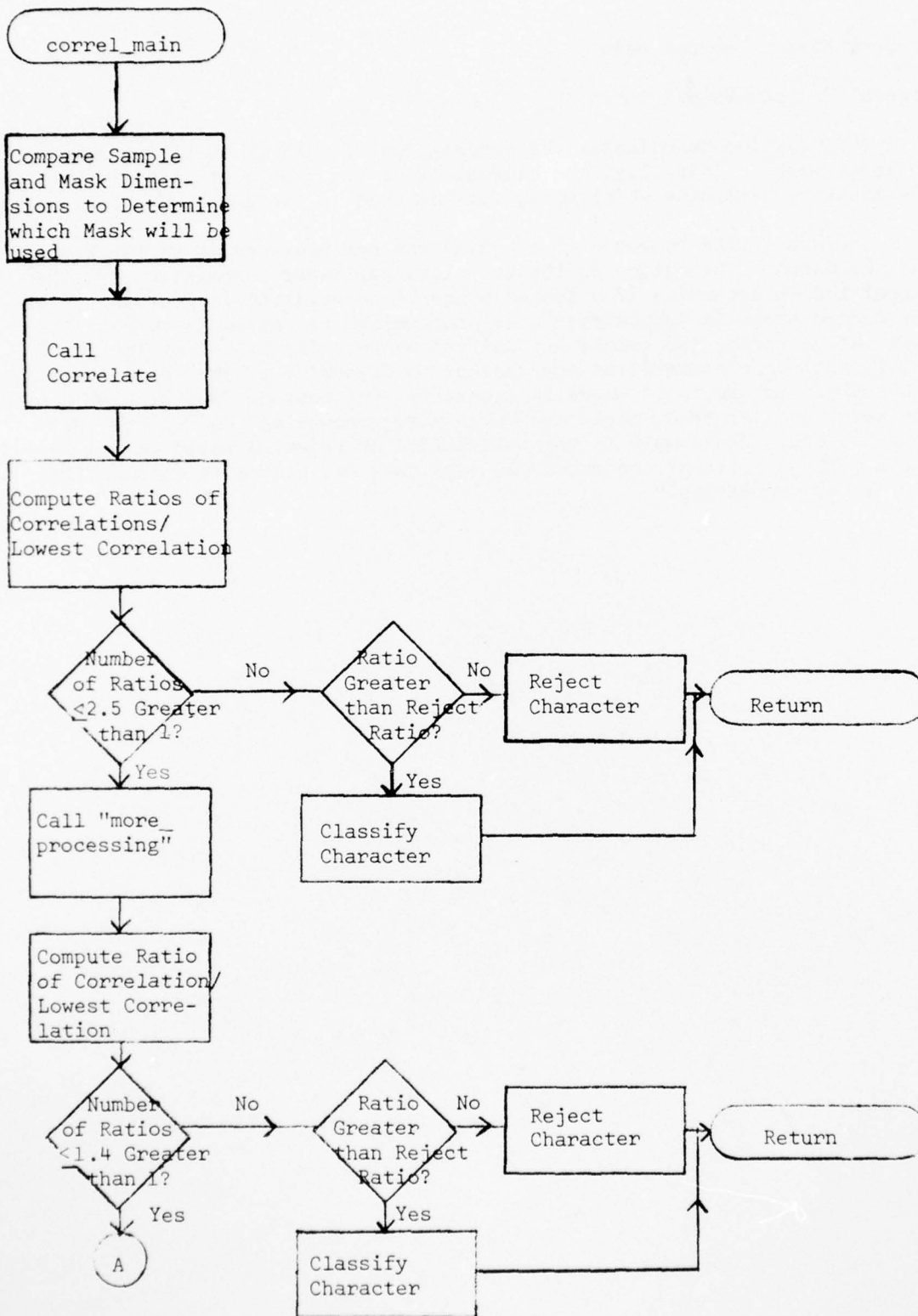


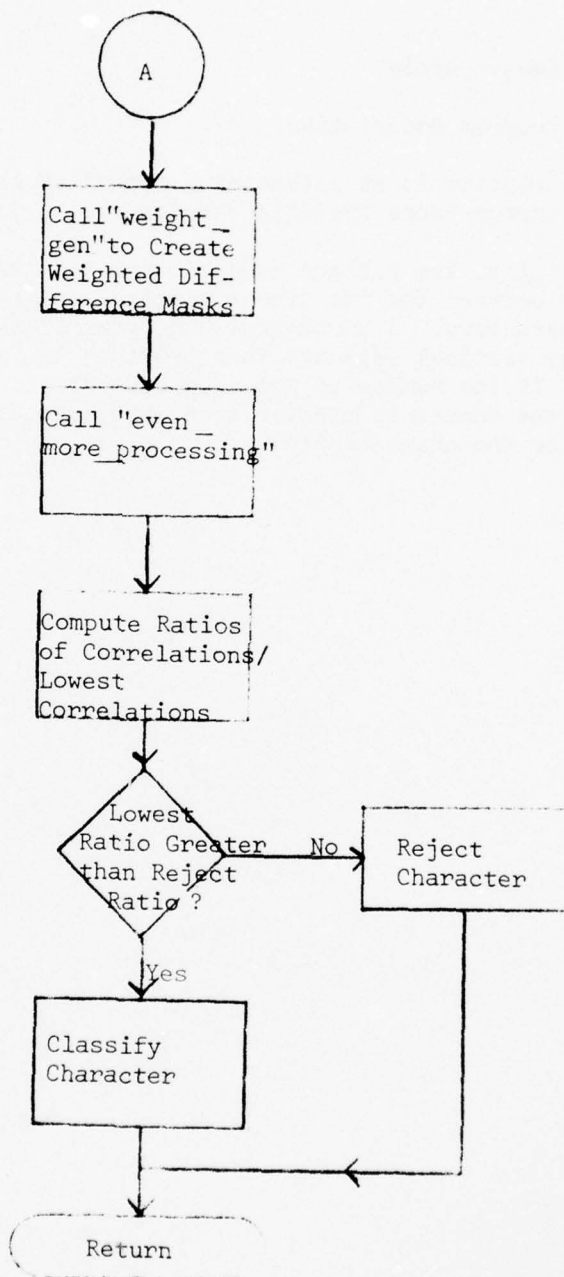
Program Name: correl_main

General Program Description:

This routine coordinates the correlation of unknown character samples to known masks. Initially, the dimensions of the sample are compared to the masks to determine which masks will be used in the correlation.

The subroutine "correlate" computes the correlations among the sample and the masks. The ratios of the correlations/lowest correlation are then calculated to determine if a second stage of correlation is necessary. If the second stage is necessary, "more_processing" is called to compute the correlations among the sample and shifted masks. The ratios of the correlations/lowest correlation are checked to determine if the third stage is necessary. If the third stage is necessary, the routine "weight_gen" creates the weighted difference masks and "even_more_processing" calculates the correlations. The sample is then classified or rejected based on the reject ratio. The results of the processed data sets are placed in output files for use by "summarize."



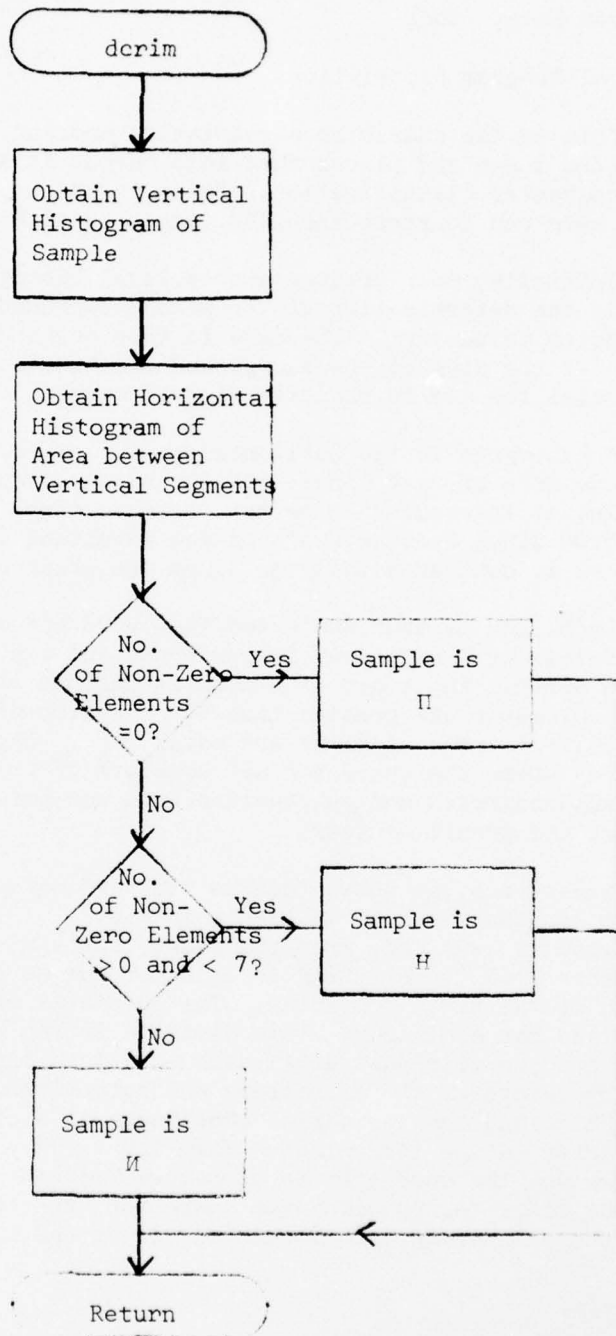


Program Name: dcrim

General Program Description:

This routine is an alternative method of logic to be used to classify the most error-prone Cyrillic characters, Н, И and П.

Initially, the pattern is subjected to a vertical histogram to isolate the area between the two strong vertical segments contained in each of the three characters. A horizontal histogram of the pixels located between the two large vertical segments then provides insight into the identity of the sample. If the number of non-zero elements is zero then the character is П. If the number is greater than zero but less than seven, the identity is Н, else the character is И.



Program Name: docr

General Program Description:

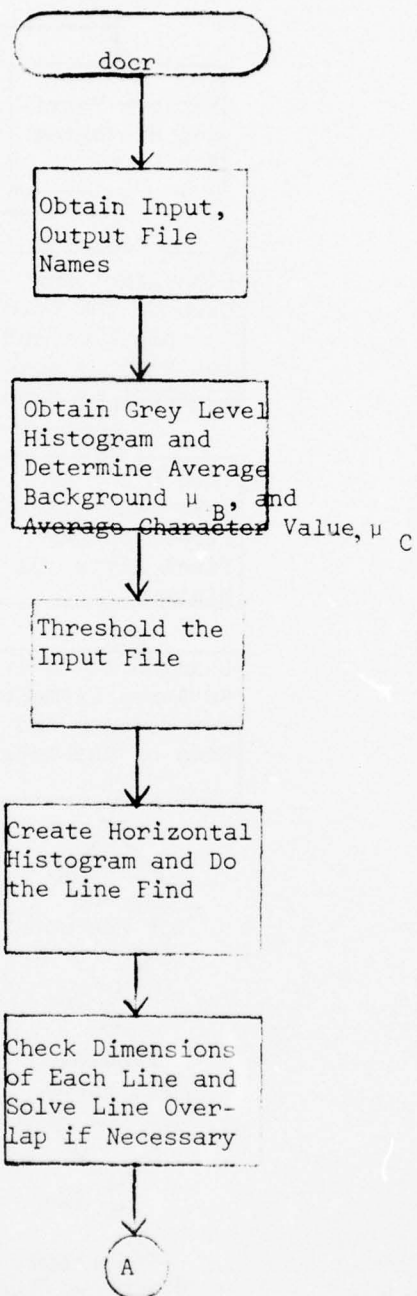
This is the character segmentation program. It extracts characters from the LIPS image and places them into output files for use in mask generation and character classification. Prior to executing docr, the program ocr will have been run to store the LIPS images as disk files.

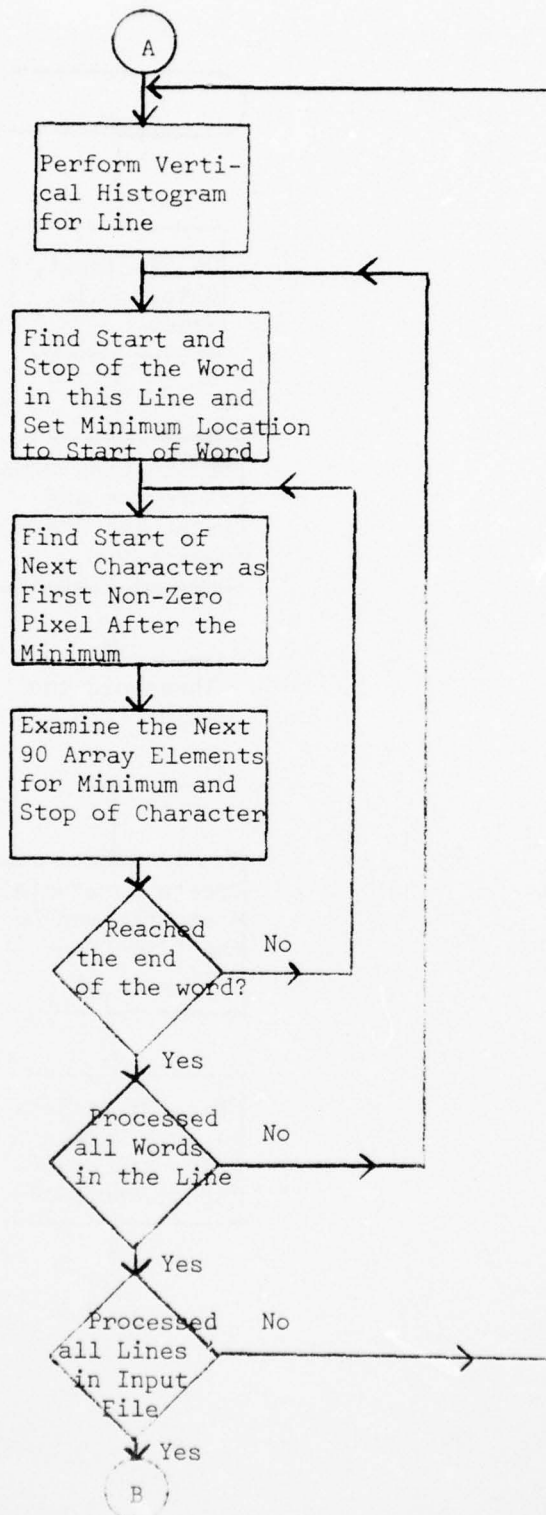
Initially, docr creates a grey level histogram from the input file for use in the determination of the mean background value, μ_B , and the mean character value, μ_C . The data is then thresholded at a value estimated to be at the stop of the background distribution. The thresholding is essential for use in the line find algorithm.

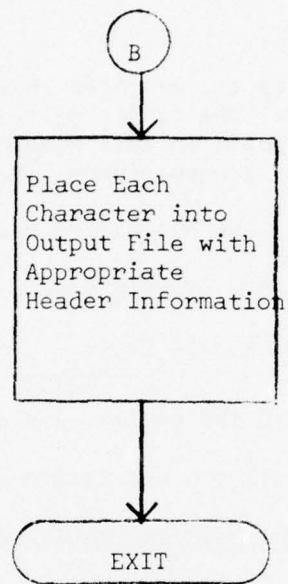
A histogram in the horizontal or X direction is calculated and examined for presence of text lines. If the width of any line is greater than 130 pixels, it is assumed to be two or more lines. This "line" is then segmented into two lines and the width of the resulting lines are examined. This process is continued until no lines are greater than 130 pixels in width.

Each line is then subjected to a word and character extraction procedure. A vertical or Y histogram is performed and the column summations are stored in an array. The start of a word is defined as that point where 12/15 array elements are greater than 0. The stop of a word is defined as the point where 12/15 array elements are equal to 0. Each word is examined for "characters" where the character set consists of printable characters; such as the alphanumerics and punctuation, and non-printable characters such as ink smears and pencil smudges.

Experience has shown that no standard characters in the Cyrillic or Latin alphabets exceed 90 pixels in width. Initially, the array representing the word is examined, from the start, for a minimum over 90 pixels. The distance from the starting location of the word to the first minimum is the width of the first character. The 90 pixels after the first minimum are examined for a minimum. This distance is the width of the second character. This process continues until each word in a line is examined and the coordinates for the width of the characters are determined. Given the vertical coordinates of a line and the horizontal coordinate of a character, the height of each character can be determined. When the x and y dimensions of each character are known, the character is extracted from the input file and placed into the output file. Along with each character, the font, and the date are extracted and the dimensions are stored for use by the classification programs.







Program Name: even_more_processing

General Program Description:

This routine calculates the weighted correlation between the input character X and the mask M. The Fisher pairwise logic is used to select the mask M as the id.of the input X in this routine. For K given masks, M_1, \dots, M_K , there are $K(K-1)/2$ comparisons: M_1 vs M_2, \dots, M_1 vs $M_K, \dots,$

M_{K-1} vs M_K . The weights w_i are calculated based on the difference between

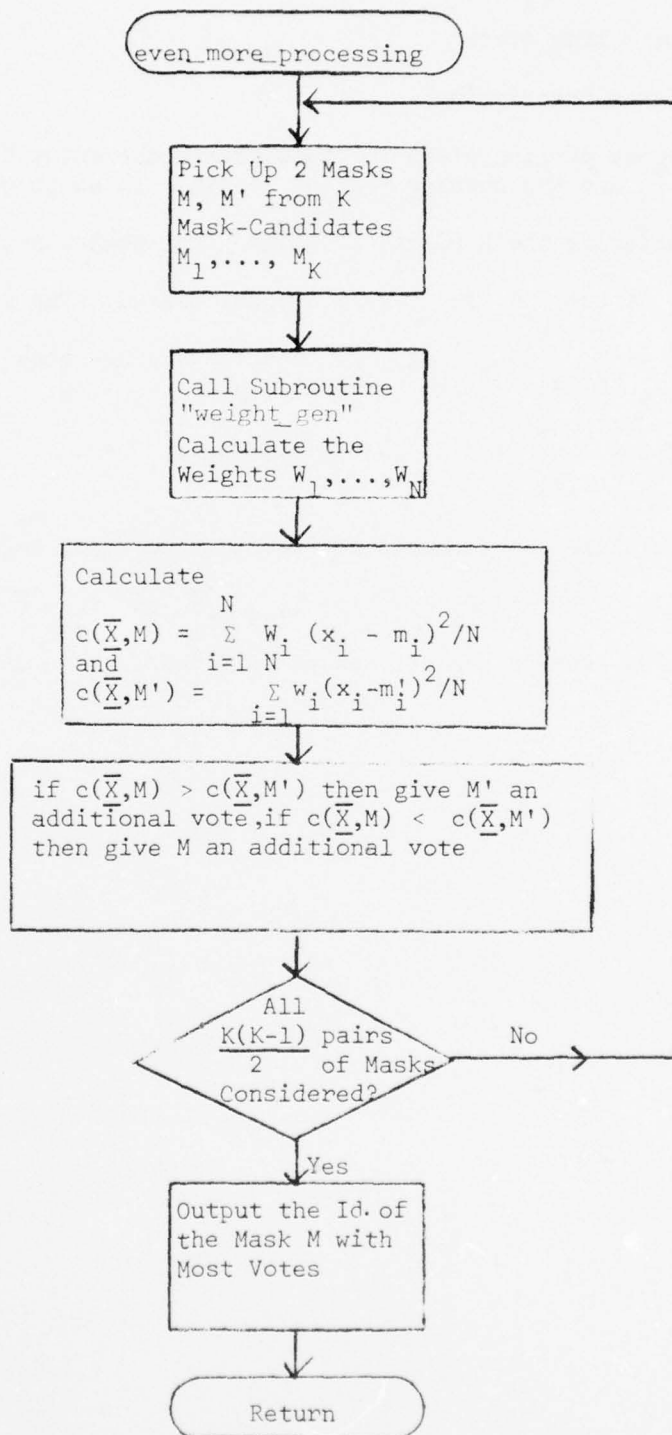
the two involving masks M, M'. For instance, $w_i = |m_i - m'_i|$. Then the

two weighted correlations $c(X,M) = \frac{\sum_i (x_i - m_i)^2 w_i}{N}$ and $c(X,M') = \frac{\sum_i (x_i - m'_i)^2 w_i}{N}$

are compared. The mask with the smaller correlation value will get an

additional "vote". After all the $K(K-1)/2$ comparisons are made, the mask

M with the most votes is assigned as the id.of the input character X.



AD-A033 537

PATTERN ANALYSIS AND RECOGNITION CORP ROME N Y
PROTOTYPE PAGE READER DESIGN STUDY. (U)

F/G 9/2

UNCLASSIFIED

NOV 76 P C KANE, E G FISHER, F H FENG
PAR-76-22

RADC-TR-76-325

F30602-75-C-0269

NL

2 of 2
ADA033537



END

DATE
FILMED
2 - 77

Program Name: line_stats

General Program Description:

The purpose of line_stats is to calculate the entropy E , the average grey level \bar{f} , and the average segment length \bar{L} in an image line.

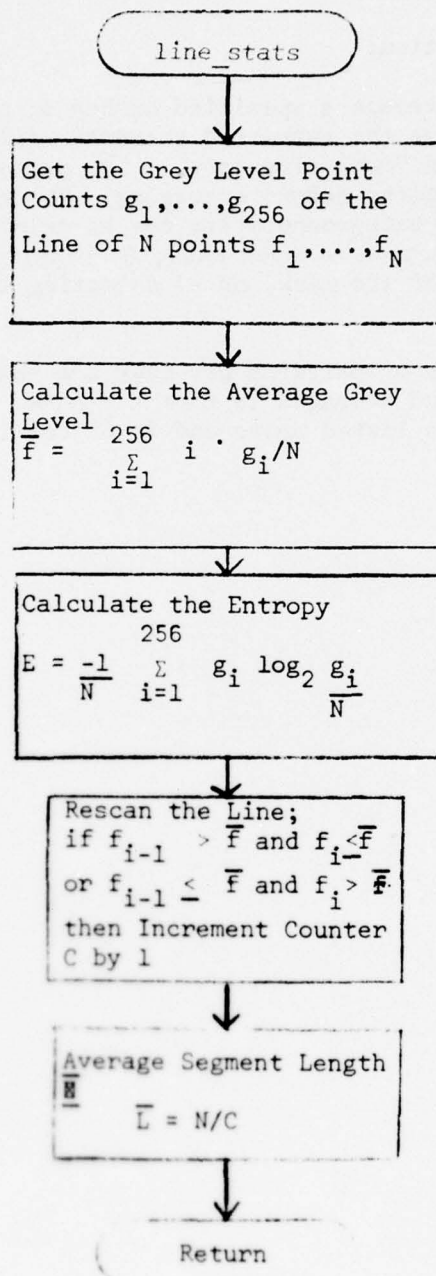
After entering the N points f_1, \dots, f_N in a scan-line, the b point-counts g_1, \dots, g_{256} of the 256 grey levels are calculated. The entropy

$$E = -\frac{1}{N} \sum_{i=1}^{256} g_i \log \frac{g_i}{N} \text{ and the average grey level}$$

$$\bar{f} = \frac{1}{N} \sum_{i=1}^{256} i \cdot g_i \text{ are calculated.}$$

The input line is rescanned to find the number of segments C as follows:

if $f_{i-1} > \bar{f}$ and $f_i \leq \bar{f}$ or $f_{i-1} \leq \bar{f}$ and $f_i > \bar{f}$ then a new segment is found. The average segment length is $\bar{L} = N/C$.

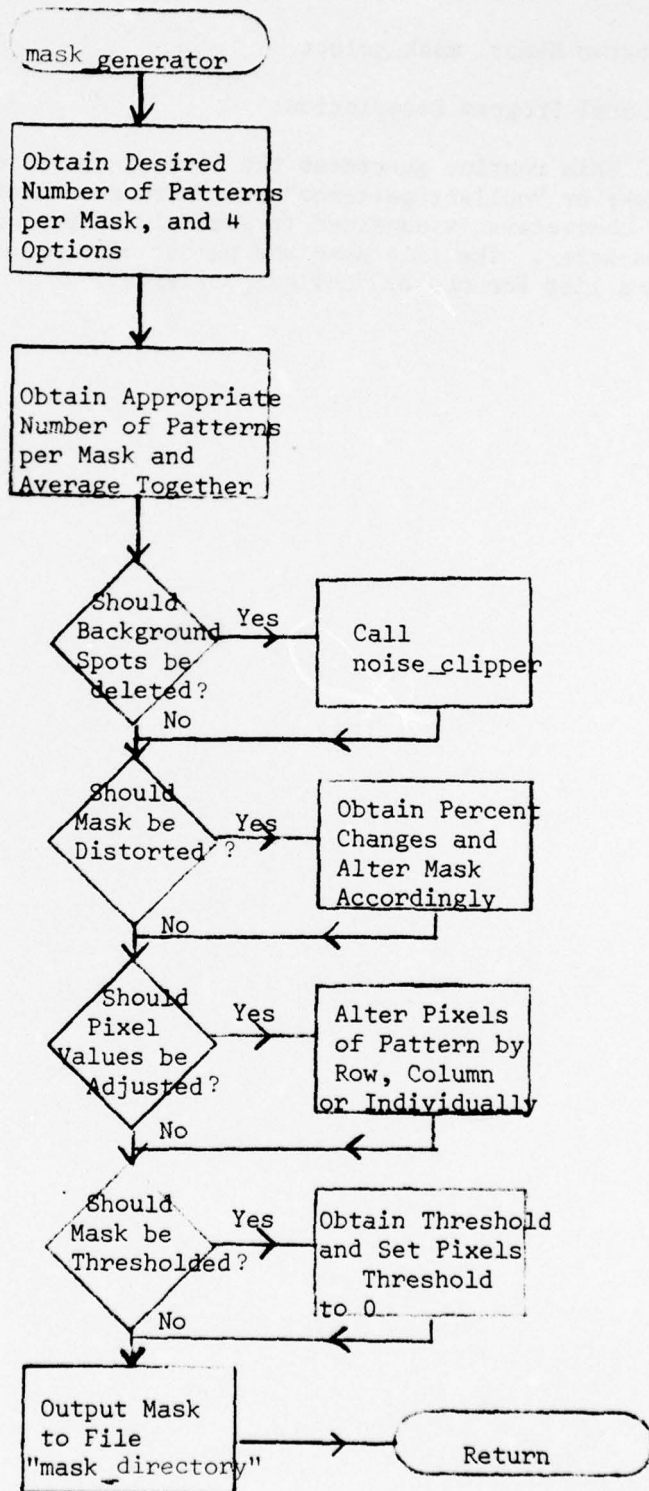


Program Name: mask_generator

General Program Description:

This program will average a specified number of patterns to form a mask. The input data are the extracted character files and the output data are masks stored in "mask_directory". The pattern centroids are used for registering the patterns before averaging. There are four options available: 1) isolated background spots may be deleted before averaging, 2) two-dimensional distortion of each mask, 3) pixel values can be adjusted thus blacking out part of the mask, or 4) selecting a threshold value for the mask.

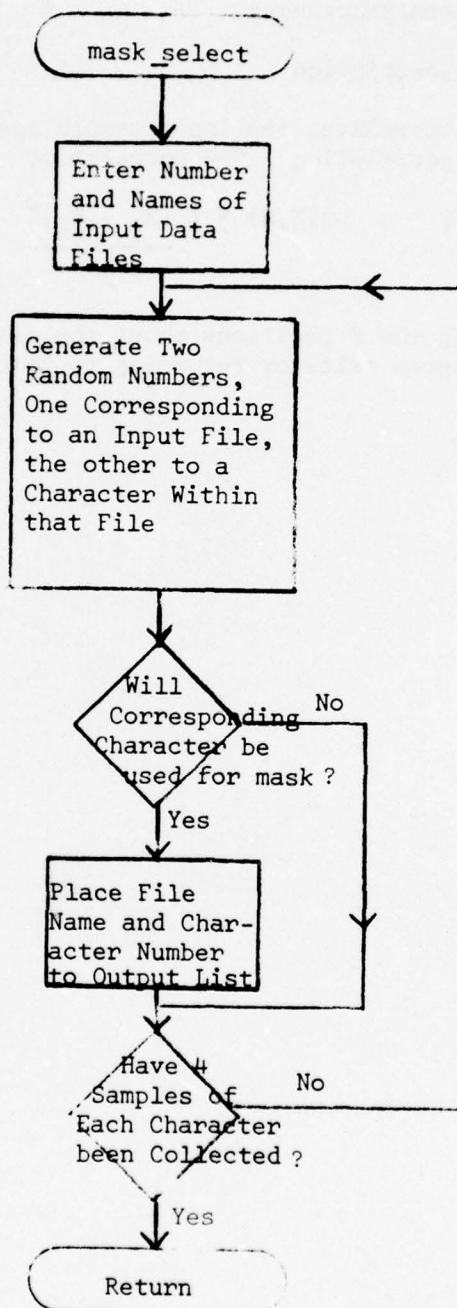
The specified number of patterns per mask are then extracted from the input file "mask_set" and averaged to form the mask. This mask is then subjected to the options listed above and is placed in the output file "mask_directory".



Program Name: mask_select

General Program Description:

This routine generates the list of characters to be used in creating masks by "collect_patterns" and "mask_generator". The identification numbers of characters is assessed to randomly select, when possible, 4 samples of character. The file name and number of sample to be used is then output to a list for use by "collect_patterns".



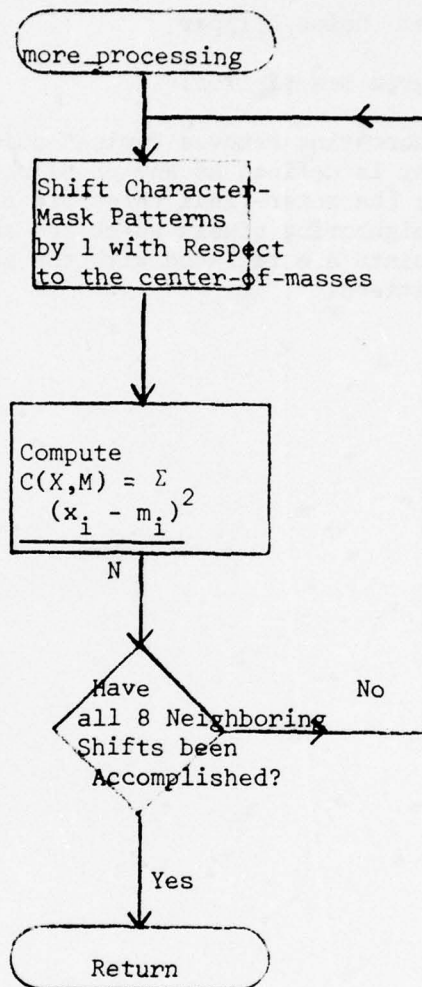
Program Name: more_processing

General Program Description:

This routine correlates the input sample against shifted masks in the second stage of correlation. The correlation

$$C(X,M) = \frac{\sum (X_i - M_i)^2}{N}$$

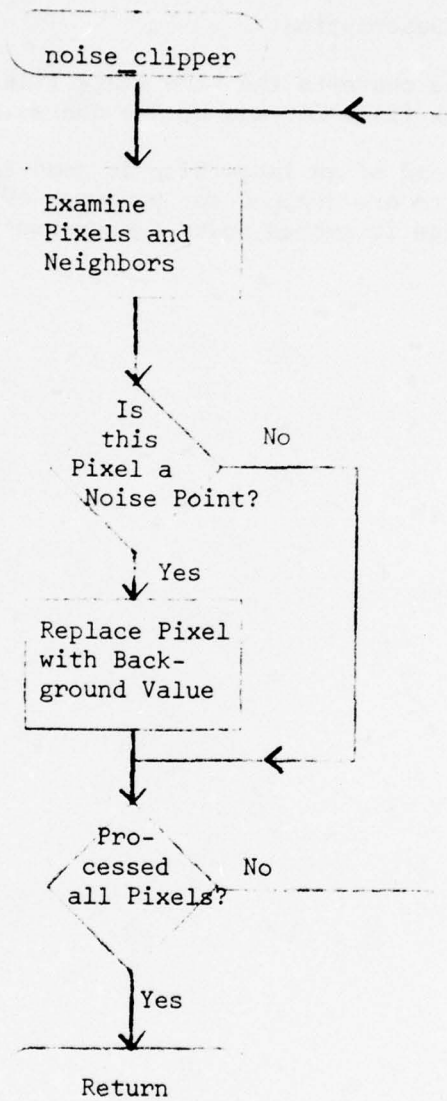
is computed using the 8 positions about the mask center-of-mass as center points. The program exits by returning to "correl_main".



Program Name: noise_clipper

General Program Description:

This subroutine removes "noise" points from a character pattern. A noise point is defined as any pixel whose grey-level intensity is above the character-final threshold but which does not have at least two neighboring pixels which are above that threshold. All noise points are replaced with the background value of the character pattern.

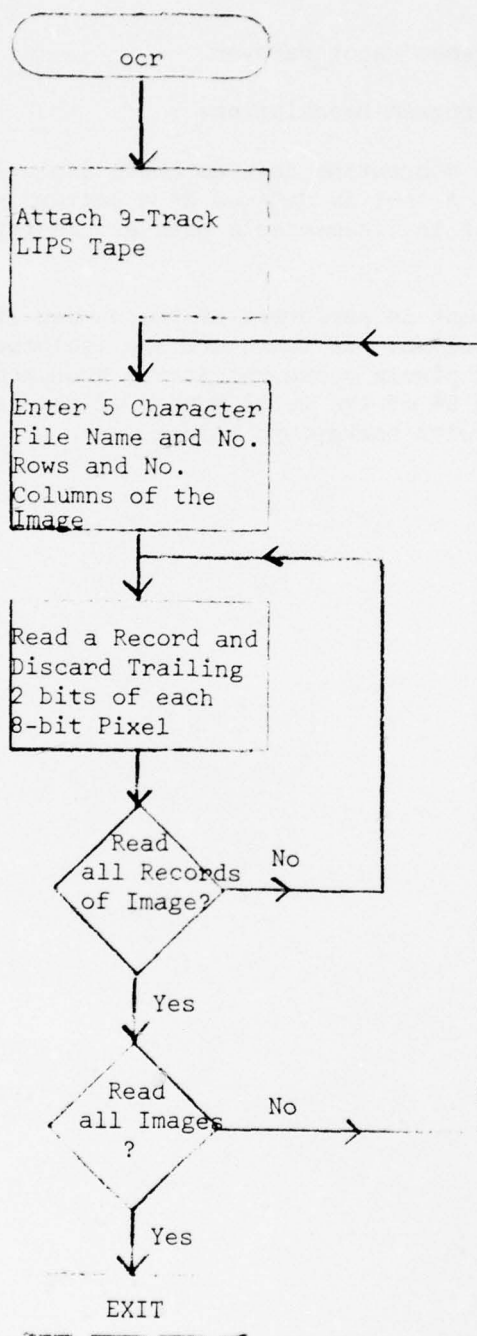


Program Name: ocr

General Program Description:

This routine converts the LIPS image files from 9-track magnetic tape into MULTICS disk files for use by the character segmentation program, docr.

As each record of an image file is read into MULTICS, the last 2 bits of each 8-bit byte are dropped for purposes of minimizing storage requirements. Each image is stored under a user-supplied unique 5-character file name.

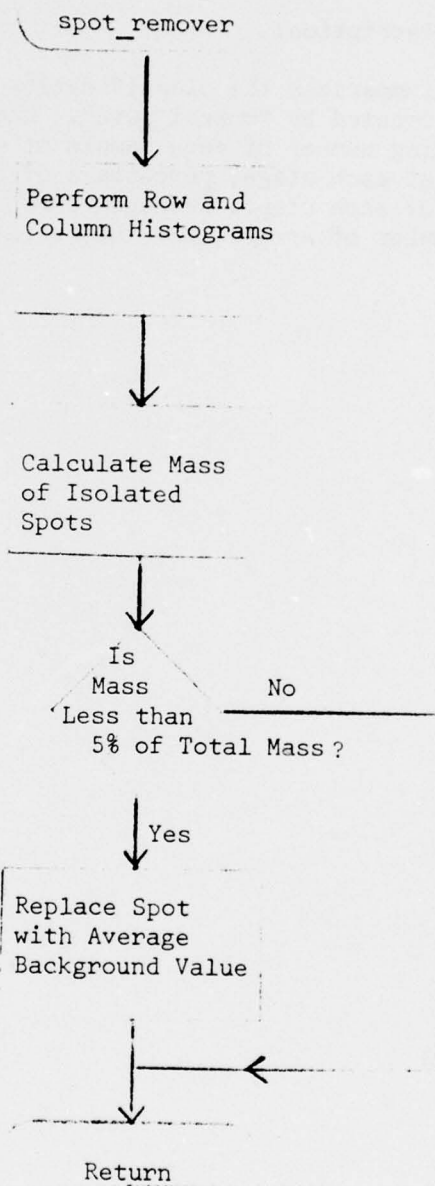


Program Name: spot_remove

General Program Description:

This subroutine deletes small isolated "spots" from a character pattern. A spot is defined as a pattern whose total mass is less than 5% of the character's mass and is totally disjoint from the pattern.

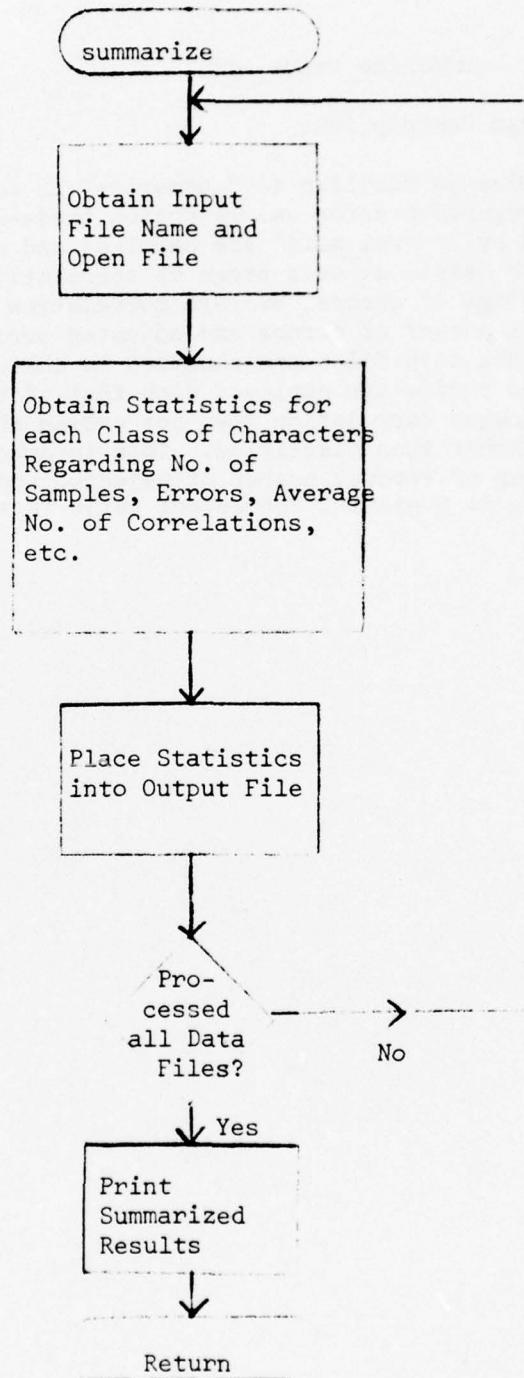
A count is performed of the number of non-zero pixels in each row and column. If there are any isolated rows and columns, the total number of pixels contained in the area are summed. If this figure is less than 5% of the total number of non-zero pixels, this area is replaced with background values.



Program Name: summarize

General Program Description:

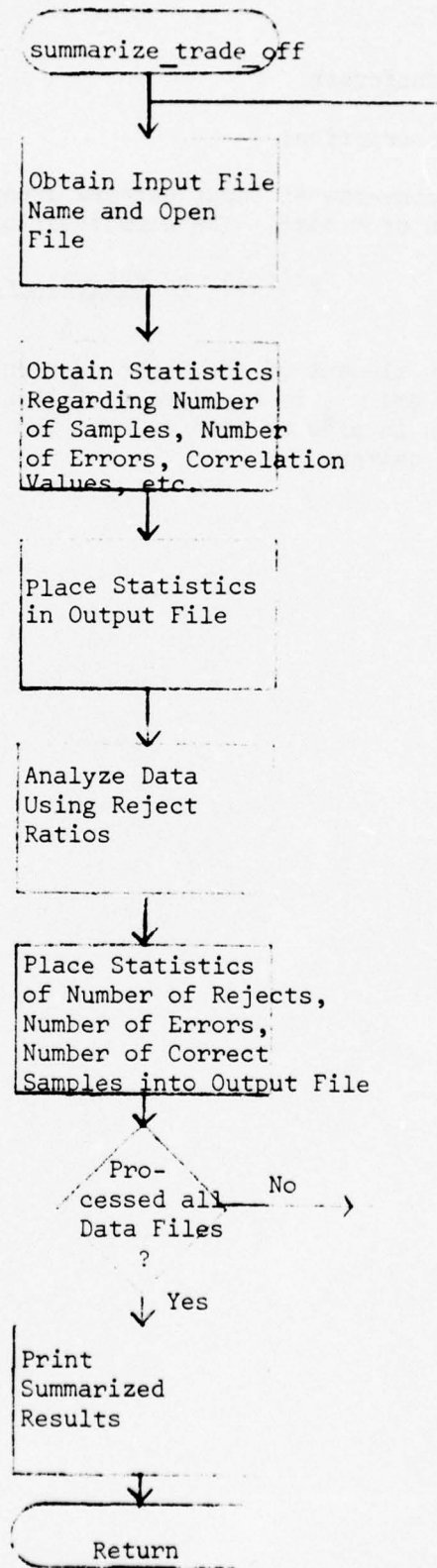
This program summarizes the classification results that are stored in the output files created by "correl_main". Each output file is examined and statistics regarding number of each sample at each stage of correlation, number of errors at each stage, percentage of errors, average number of correlations per sample at each stage, average correlation value, total number of samples, total number of errors, and the overall adjusted error rate are presented.



Program Name: `summarize_trade_off`

General Program Description:

This routine is similiar to "summarize" in function but also presents information regarding error vs. rejection trade-off rates. The output files created by "correl_main" are examined and statistics describing number of each sample at each stage of correlation, number of errors at each stage, percentage of errors, average correlation value, total number of samples, total number of errors and adjusted error rate are presented. In addition, the data files are analyzed to observe the effect of reject ratios. These ratios are employed such that if the ratio of second lowest correlation/lowest correlation does not exceed the reject ratio, the sample is rejected rather than classified. This information relating the number of samples, number of errors, number of rejects, and number of correctly classified samples is presented for reject ratio rates varying from 1.00 to 1.30.



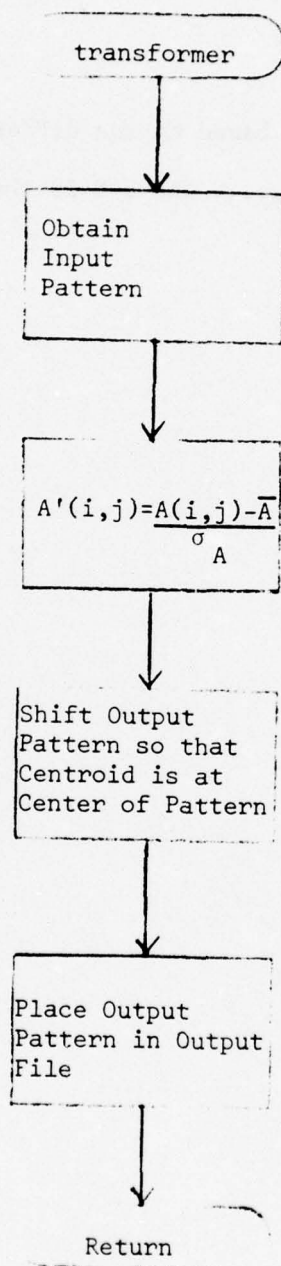
Program Name: transformer

General Program Description:

This routine converts an input pattern array of 6 bits to a grey-level normalized pattern of 4 bits. The normalization formula is

$$A'(i,j) = \frac{A(i,j) - \bar{A}}{\sigma_A}$$

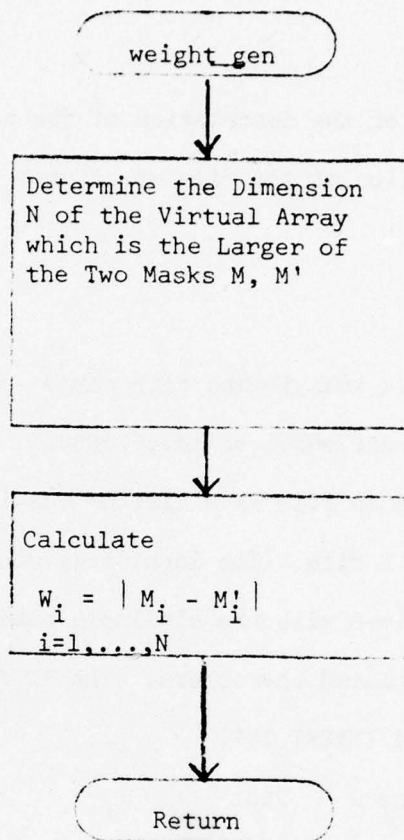
where $A(i,j)$ is an element of the input pattern, \bar{A} is mean grey-level of the input pattern and σ_A is standard deviation of the input pattern. The output pattern is also shifted so that the centroid of the pattern is located at the center.



Program Name: weight_gen

General Program Description:

Calculate the weights W_i based on the difference between two masks M, M' by $W_i = |M_i - M'_i|$, $i = 1, \dots, N$ where N is the dimension of the larger of M and M' .



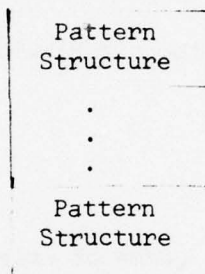
FILE STRUCTURES

An important part of the description of the software developed for this project is the description of the file structures used by the programs.

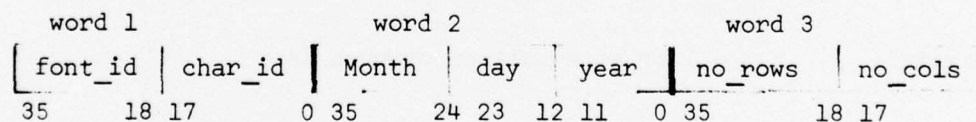
File Type:	ID
Naming Convention:	id.< transformed file name> or id. < DOCR file name>
Example:	id.c6t.c6001 or id.c6.c6001
Purpose:	The ID file is a list of the identities of the characters of a file. The identities contained in the file correspond 1-to-1 with the similarly named transformed file of isolated characters. The ID file is used by MASK_SELECT and INSERT_IDS.
Format:	line 1 list lines 2...a list of the ids of the characters in the corresponding transformed file; a zero ("0") is used wherever the corresponding character is not to be considered for classification (since such classification would be meaningless).
File Type:	DOCR
Naming Convention:	< font name> . < name of original image file>
Example:	c6.c6001

Purpose: The DOCR file (usually output by DOCR) contains several (1 or more) individual, isolated character images. DOCR files are also used for masks (output by MASK_GENERATOR) or collections of selected patterns (output by COLLECT_PATTERNS).

Format: 1 or more pattern structures



where a pattern structure consists of a three-word header (36 bit words) followed by the pattern data array.



The header words are:

font_id: Identity of the original document from which the data was taken.

char_id: Identity of the character pattern, if known.

month,day,year: The date of isolation of the sample.

no_rows: The number of rows of data in the pattern.

no_cols: The number of columns of data in the pattern.

The pattern data array is laid out as no_rows lines of data. Each line of data is a whole number of MULTICS words (36 bits) packed with no_cols 6-bit intensity values and padded (on the right) with zeroes.

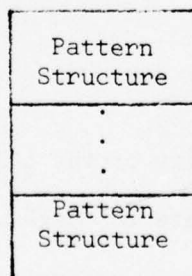
File Type: TRANSFORMER

Naming Convention: .t. \name of original image file>

Example: c6t.c6001

Purpose: The TRANSFORMER file (output by OUTPUT_MULTICS, a subroutine of TRANSFORMER) contains a collection of characters to be input by the correlation program CORREL_MAIN.

Format: 1 or more pattern structures



where a pattern structure consists of a 3 word header, a fourth word which provides the fill word, and a pattern

array. The three-word header contains 12 seven-bit fields which represent:

font_id: The identity of the document from which the pattern was extracted.

char_id: The identity of the character contained in the pattern array.

month,day,year: The date of pattern extraction.

no_rows, no_cols: The virtual size of the pattern array.

size: 1

hi_row: The highest row coordinate of actual pattern data within the virtual array.

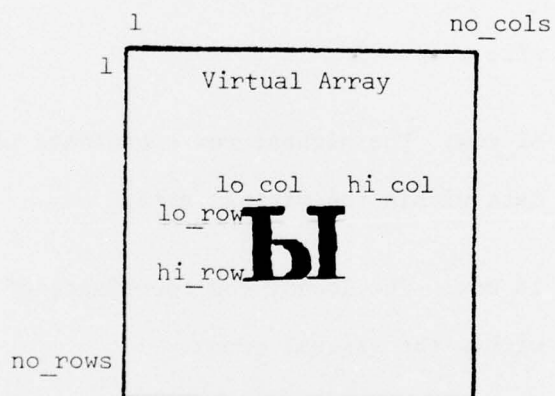
lo_row: The lowest row coordinate of actual pattern data within the virtual array.

hi_col, lo_col: The analogs of hi_row and lo_row with respect to the column coordinates.

The fill word is that particular normalized intensity which is to be assigned to each value in the virtual array which is not in the actual array.

The pattern array data consists of a continuous string of normalized intensity values for the actual pattern array. The number of bits of normalized intensity is specified by the variable "precision" in the file MASK_DIRECTORY."

Pictorially the arrangement may be viewed as:



APPENDIX B
SINGLE CHARACTER PROCESSOR TIMING

APPENDIX B

SINGLE CHARACTER PROCESSOR TIMING

To better understand the processing time requirements for the character processors, reference should be made to Figure 6-1 for an overview of the hardware configuration, and Figure 3-9 in Section 3, a chart of the system data flow.

The microprocessor used for these timings has a cycle time of 150ns. Addition and subtraction of sixteen bit words requires three cycles with multiplication completed in two cycles using a table look-up technique.

The character processor (CP) system timing is based on a 32-character font with the characteristics outlined in Figure B-1.

The timing is primarily concerned with the arithmetic function of the correlations between the incoming data and the stored character masks, with a 50% factor added for data handling overhead.

To facilitate the timing calculations, the average size mask, expressed in pixels, and the average number of correlations performed per character for each of the three correlation phases were computed. When calculating the above averages, the frequency of occurrence for each character was used as a weighting factor.

The algorithm for determining the average mask size for each of the three phases then became:

$$AMS = \frac{1}{ANC} \sum_{n=1}^N f(n) \cdot M(n) \cdot R(n) \cdot C(n)$$

where

- $f(n)$ = Frequency of occurrence of the n^{th} character (expressed as a fraction)
- $M(n)$ = Number of masks used (masks within $\pm 15\%$ of n^{th} char. size)
- $R(n)$ = Number of pixel rows
- $C(n)$ = Number of pixel columns
- N = Number of characters in font
- ANC = The average number of correlations per data character:

Character	Frequency of Occurrence in Russian Text	Size in Pixels		No. of Masks used for Correlation I	No. of Masks used for Correlation II	No. of Masks used for Correlation III
		No. Rows	No. Columns			
а	7.30%	59	47	17	4	3
б	1.57	83	47	5		
в	4.61	51	44	17	4	2
г	1.33	53	39	13	3	
д	2.75	67	51	6	2	
е	9.66	55	43	13	4	2
ж	0.81	57	73	6		
з	1.68	51	37	10	2	
и	8.88	51	55	11	4	3
й	0.93	87	57	5	2	
к	3.54	59	51	13	6	5
л	4.55	57	47	13	8	
м	3.25	57	60	9	6	
н	6.67	57	55	12	4	3
о	10.47	47	42	18	5	
п	2.40	51	55	11	5	3
р	5.27	71	51	6		
с	5.37	59	41	13	3	
т	6.64	55	41	16	2	2
у	2.53	71	50	5		
ф	0.31	90	63	2		
х	1.03	49	53	13		
ц	0.31	67	53	5	2	
ч	1.33	53	51	13	7	
ш	0.81	55	79	2	2	
щ	0.41	73	77	7		
ъ	0.01	55	57	11		
ы	2.45	53	65	5	3	
ь	1.19	51	45	18	5	2
э	0.37	53	41	10	5	
ю	0.49	51	67	4		
я	1.07	57	53	12	4	

Figure B-1 Characteristics of Typical Font

$$ANC = \sum_{n=1}^N f(n) \cdot M(n) \quad \text{Phase 1}$$

$$ANC = 4 \sum_{n=1}^N f(n) \cdot M(n) \quad \text{Phase 2}$$

NOTE: The multiplier of 4 represents four correlations made on each character with the centroid point shifted one pixel distance from center in 4 positions 90° apart, performed on the average in Phase 2.

The operations required to complete a character correlation consist of a subtraction, an addition, and a multiplication for each pixel and one division per character. These operations are derived from the correlation algorithm (See Section 3).

The time to complete the correlations in each phase is expressed in the following algorithm.

$$(AMS \cdot T_p \cdot ANC \cdot 1.5) + 750 = T_C$$

where

- AMS = Average mask size
- T_p = Calculation time for one pixel
- ANC = Average number of correlations per character
- 1.5 = 50% data handling overhead
- 750 = Division time in nanoseconds
- T_C = Correlation time per character in nanoseconds

Correlation time per character is calculated as follows:

Phase 1: $2840 \times 1200 \times 12.4 \times 1.5 + 750 = 63,389,550\text{ns} = 63.4\text{MS}$

Phase 2: $2875 \times 1200 \times 7.9 \times 1.5 + 750 = 40,883,250\text{ns} = 40.9\text{MS}$

Phase 3: $2897 \times 1200 \times 0.4 \times 1.5 + 750 = 2,086,590\text{ns} = 2.1\text{MS}$

Total correlation time per character = 106.4MS

Further investigation will be necessary to determine the exact heuristic procedures that will provide the optimum results in Phase 3 of the character processing. A pixel weighting scheme, as well as empirical measurements for individual characters, is to be evaluated. Final selection of the Phase 3 procedure could result in a variance of as much as minus 50% to plus 100% of the Phase 3 timing.

APPENDIX C

DATA SAMPLES

APPENDIX C

DATA SAMPLES

The Cyrillic journals from which the data for the project was extracted are listed below. In the instances where the identifying font number is the same for both languages, such as L1 and C1, the Latin is the translation of the Cyrillic journal and is located in the same journal.

Samples from each of the fonts are located below and contain both the original chips as digitized by the LIPS system and MULTICS representation of the character.

Font: C1, L1	"Нейрофизиология"
C2, L2	"Журнал высшей нервной деятельности"
C4, L4	"Журнал микробиологии, эпидемиологии и иммунологии"
L5	"Коллоидный Журнал"
C6	"автоматика и телемеханика"
C8, L8	"Журнал экспериментальной и теоретической физики"

current ($1 \cdot 10^{-6}$ — $6 \cdot 10^{-9}$ A) was passed by means of a barrel microelectrode inserted in a horizontal cell of the turtle retina. The effect on the retina was accompanied by an increase in resistance of the microelectrode (M). This increase resulted in a change of the cell length response, so that the changes in resistance are localized not in the cell membrane but in the cell near the microelectrode tip. The effect described can be reproduced by passing current through the second barrel of a double-barreled microelectrode. It had to be several times stronger, than in passing through the first barrel. The membrane potential of a horizontal cell was shifted to a more negative potential (by means of a constant current passed through the cell). The effect of the microelectrode resistance increase during hyperpolarization is simultaneously with hyperpolarization light response itself. On the other hand, extrinsic hyperpolarization of the cell membrane was accompanied by a decrease in the microelectrode resistance. The effect found can be explained by the "closing up" of the microelectrode intracellular structure, the resistance of which is a function of the

ylcyclase and produces

Font L1 Sample Data

istics of evoked activity to light were studied in the visual cortex of the human. The results of the study of the reactivity asymmetry were found. The first is predominant at binocular stimulation and is linked with a greater amplitude of P4 and of sensory alpha-averaged discharge of the subdominant hemisphere. The second type of asymmetry is observed in cases of monocular stimulation. It consists in a bilateral P3-P4 oscillations (sometimes the P2-P2 complex is observed). The effect of the stimulation of the right eye. This effect is related to the anatomical reinforcing retino-prestriate pathway which affects the structures of the temporal lobe of the right (subdominant) hemisphere as its major link.

ealed in the unitatio

Font L2 Sample Data

C-2

Activation of allergic focal reaction under the effect of sp
pigs with dysentery keratoconjunctivitis was accompanied by
dular and motor apparatuses of the intestine, and disturbance of
Conjunctival application of a complete Sonnei antigen and Troitsk
se of alkaline phosphatase secretion (1.34—1.39-fold), and of ente

Patients with acute and chronic enterocolitis, acute and
marked functional disturbances of the glandular apparatus and
vity of the intestine reflecting the pathogenetic regularities in
fection of the intestinal tissue of specific and nonspecific or

Indices of enteropeptidase and alkaline phosphatase acti
could serve as reliable diagnostic criteria of functional disturb
and chronic inflammatory diseases of the intestine of infection
and, along with results of other methods of investigation, aide
sentry from carrier state.

xicsubphagocytosisandp

Font L4 Sample Data

A study has been made of the structure of the blends
amide and polyethylene with polyamide AK 60/40 form
of unstabilized and commercial polymers. Depending c
a melt, it is possible to obtain solid dispersions polymer
ture or disperse systems of interpenetrating macronet
Small amounts of surfactants favor the formation of fil
formation accompanied by «necking» breaks up into separa

ynthesisofcalciumionk

Font L5 Sample Data

Instability of an electron beam in a low concentrated experimentally. A special feature of the experiment beam ($23 \cdot 10^{-2}$ cm), this being an extreme condition for the formation of plasma instabilities of the drift type. The phenomenon, which appears on development of instability, is experimental results with the theory shows that, under investigation, escape of ions from the beam is due to their field by axial-nonsymmetric electron-ion oscillations; the consecutive kinetic and hydrodynamic build-up of drift

of (equifield) which are in

8

Font L8 Sample Data

итного нерва, по-видимому, нет общих синаптической депрессия отсутствует. Графики, посредних данных, полученных в этих опытах,

ожных отраженных влияний со стороны нервной системы при раздражении дорсальных влияний, которые могут возникнуть вследствие ающего тока на противоположную сторону со ого дорсального канатика, ряд опытов провед х животных с применением глубокой двусто-ных канатиков ниже и выше места раздра-менения ПДП, вызванных пробными раздра-ались от тех, которые были обнаружены при дорсального столба.

пытов исследовалось влияние предварительного канатика на величину антидромных ответом терминалей, входящих в сегмент L₇ пер-стрируемых в афферентном нерве или фила-

ТЕНСИВНОСТЯХПРИВОД

Font C1 Sample Data

C-4

между следовыми процессами. Исследования К. С. Абуладзе [1, 2] показано, что скрытые (латентные очаги) могут проявлять доминантные свойства к вновь возникающему возбуждению. Следовым, усиливаясь за счет приходящего возбуждения, с на характер протекания поведенческой реакции. Вот ь, сигнализирующий оборонительную реакцию, пр авершения пищевого рефлекса, вызывает одновременно так же пищевой сигнал, испытанный после обор са, сопровождается бинарным рефлексом. , что после действия сильного раздражителя, тако ий ток, особенно длительно сохраняется след возбу то характера [3]. Поэтому возбуждение, возникающ щего сигнала, может полностью отвлекаться в ст ьно примененного оборонительного рефлекса. Внеи подавлением или, вернее, не проявлением пищевой в мере повторения пищевого сигнала с подкрепление вьшения возбудимости в структурах пищевого ре в обе условные реакции а затем происходит и отлич

овопросасущследую

Font C2 Sample Data

ческого иммунитета отмечается на 10-й день после стафилококковым анатоксином и проходит к 30—близительно в эти сроки (21—25-й день), т. е. в фаз телей общей реактивности, мы и обследовали наш

Таким образом, меньшая продукция антител в ответ на введение повышенных доз антигена (стаф на) явилась следствием угнетения общей реактив тверждает также дифференциальный анализ резул разном состоянии общей реактивности, свидетельс порциональной зависимости специфической реакт ской. У больных с высоким уровнем общей реа титр стафилококкового антитоксина равнялся 5,2 уровне общей реактивности он составлял $1,9 \pm 0,2$ ные позволяют сделать следующий вывод: чем низ тивности, тем меньше выражена способность о специфические антитела при иммунизации.

Нужно отметить, что полученное в результате 1 уровня антистафилококковых антител сохранялось

41 На 62 компонента

Font C4 Sample Data
C-5

то и в соответствии с таблицей случайных чисел, что в силу эргодичности процессов $x(t)$ и $y(t)$ асимптотически «временных» характеристик совпадают для всех начальных условий с вероятностью 1, и для всех начальных условий с вероятностью 1 роль «нулей» операционных усилителей во время эксперимента через каждые 30 мин., причем «непрерывно» печатались установкой соответствующих начальных условий. По полученным реализациям были вычислены среднеквадратичные отклонения и нормированные значения. Во всех случаях относительные ошибки по суммарным значениям не превосходили 5%, что соответствует вычислениям на АВМ МН-7.

Проверка гипотезы о равномерном распределении $x(t)$ проводилась по критерию Пирсона χ^2 при доверительном уровне 0,95, а проверка «случайности» сигнала $y(t)$ проводилась по критерию.

Кроме того, мультипликативный итерационный алгоритм испытывался на аналоговой части гибридной вычислительной системы ГВС-100 в Институте проблем управления. Схема генератора, содержащая 2 блока слежения (хранения) и 2 блока вычисления, была реализована на АВМ МН-7.

ркогообъектапосред

Font C6 Sample Data

в настоящее время имеется большое количество работ (см. обзор [1]) по изучению ядерного (ЯМР) и электронного (ЭПР) магнитного резонанса в жидкостях и скорости спин-решеточной релаксации (СРРЖ) в жидкостях. В статистическом описании этих экспериментов, как правило, рассматриваются отдельные молекулы и почти не учитываются коллективные эффекты в жидкости. Например, всегда считается [2], что в жидкости тот же, что и в покое. При учете влияния внутримолекулярных полей в СРРЖ рассматривается диффузия только индивидуальных молекул. Известно, что в жидкости при наличии сдвиговых напряжений ориентации несферических молекул становится неадиабатической, что приводит к двулучепреломлению света в жидкости, текущей в магнитном поле (эффект Максвелла [3]). Показано, что это ориентационное взаимодействие молекул со сдвиговыми напряжениями, а также спин-решеточное взаимодействие приводят к изменению спектра ЯМР (числа и частоты) в текущей жидкости по сравнению с покоем. В настоящее время вычисления скорости СРРЖ. Удастся связать

почти не учитыва

Font C8 Sample Data

C-6

*MISSION
of
Rome Air Development Center*

RADC plans and conducts research, exploratory and advanced development programs in command, control, and communications (C³) activities, and in the C³ areas of information sciences and intelligence. The principal technical mission areas are communications, electromagnetic guidance and control, surveillance of ground and aerospace objects, intelligence data collection and handling, information system technology, ionospheric propagation, solid state sciences, microwave physics and electronic reliability, maintainability and compatibility.

